150 ptas.



CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

WHICH KIND OF MONITOR

Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona, y comercializado en exclusiva por Distribuidora Olimpia, S.A., Barcelona

Volumen III - Fascículo 26

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones
Asesor técnico: Francisco Martín, Jesús Nebra

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti. A. Cuevas, F. Blasco

Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.°, Barcelona-8
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador,* se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-85822-94-3 (tomo 3)
84-85822-82-X (obra completa)
Depósito Legal: B. 52/1984

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 118407
Impreso en España - Printed in Spain - Junio 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, Madrid-34.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio Blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Ferrenquín a Cruz de Candelaria, 178, Caracas, y todas sus sucursales en el interior del país.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 3371872 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Distribuidora Olimpia (Paseo de Gracia, 88, 5.°, Barcelona-8), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Distribuidora Olimpia, en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.



Eficacia educativa

El ordenador puede ser un incomparable auxiliar en la enseñanza, en especial como libro de texto interactivo

En 1980 el gobierno británico lanzó un plan destinado a impartir nociones de informática tanto en establecimientos de enseñanza primaria como secundaria. Conocido como MEP (*Microcomputers in Education Project*: plan para la utilización de microordenadores en la educación), su período de vigencia se estableció en seis años, con un presupuesto global de 21 millones de libras. Quizá sea injusto dividir esta cantidad por 25 000 (el número de escuelas de enseñanza primaria, media y secundaria sólo de Inglaterra) y suponer de este modo una distribución equitativa de los recursos. La máquina que más se recomendó fue el BBC Micro, cuyo precio en el mercado británico es de 400 libras. La segunda opción, el 380Z de Research Machines, tiene un precio bastante más alto que el citado microordenador. Dado que el presupuesto para el proyecto era a todas luces insuficiente, las escuelas se vieron obligadas a recurrir a la autofinanciación.

Hacia 1983, mediado el período de vigencia del plan, el ministro de Tecnología de la Información estuvo en condiciones de proclamar que todas las escuelas secundarias del país (4 553 en Inglaterra) tenían un ordenador, al igual que la mitad de las escuelas primarias. No obstante, la mayoría de los méritos en este sentido se deben atribuir a las asociaciones de padres de alumnos, instituciones benéficas y hasta a los propios escolares, que con gran entusiasmo realizaron considerables esfuerzos para recaudar fondos.

En vez de actualizar el plan de estudios para las escuelas y aumentar su aplicabilidad parecería que la iniciativa del gobierno ha agudizado problemas existentes. La necesidad de equipos costosos ha aumentado la frustración tanto de los maestros como de los alumnos. Ahora existe una conciencia clara de que cada vez se acentúa más la desigualdad de oportunidades entre los niños que pertenecen a ambientes familiares comparativamente más adinerados, quienes disponen de más dinero para contribuir a proyectos como éste, y aquellos niños que pertenecen a familias de menores recursos. Debido al reducido número de microordenadores de que disponen los alumnos de cualquier escuela, es poco probable que el "alumno promedio" tenga acceso a una máquina durante más de 15 minutos por semana, tiempo apenas suficiente para iniciarse en un plan de alfabetización informática y absolutamente insuficiente para explorar las posibles ven-

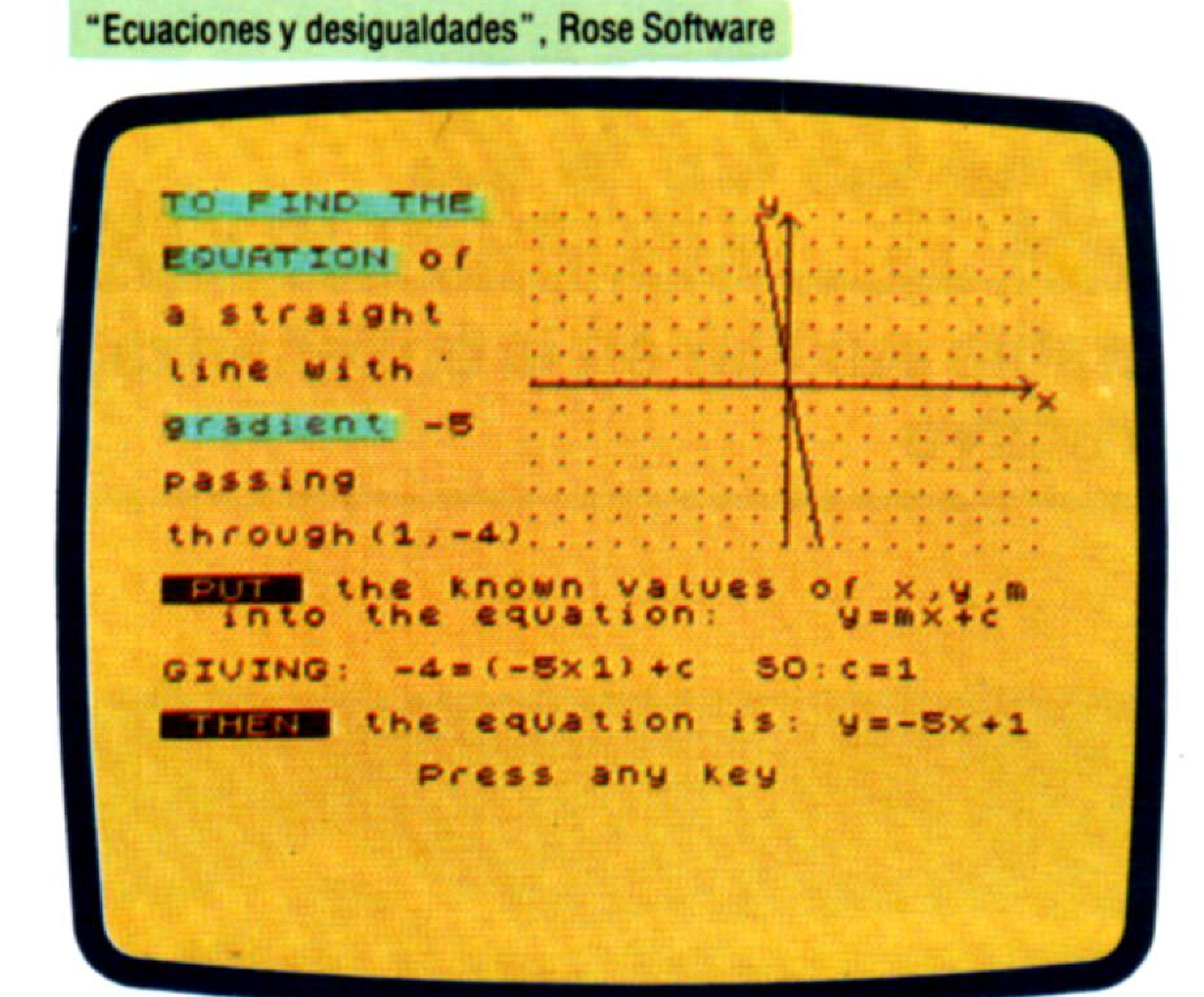


Una senda segura
La aspiración de un mejor futuro para sus hijos la expresan muchos usuarios de ordenadores personales a través de la compra de software educativo. Para los niños pequeños existe una gran riqueza de material didáctico de iniciación, y para los de mayor edad, que cursan estudios más avanzados, existe un número aún mayor de paquetes de repaso para planes de estudio muy específicos destinados a diversos microordenadores

Alan Adler

Paquetes científicos
El campo de acción de los paquetes de software creados para ayudar al estudiante a repasar asignaturas específicas es limitado. Cada uno de ellos intenta cubrir un único aspecto de la materia (matemáticas, o literatura, p. ej.) con el mayor detalle. Nuestro primer ejemplo trata sólo con ecuaciones matemáticas de distintas clases (lineales, de segundo grado y simultáneas) y es para el Spectrum, mientras que el paquete de geometría diseñado para el mismo hardware es de alcance más amplio. Los dos ejemplos restantes plantean problemas relativos a la ley de Ohm y el diseño de amplificadores

"Ecuaciones y desigualdades", Rose Software



tajas de los paquetes de software educativo interactivo.

Dada esta falta de adecuación del sistema escolar, quizá ya no resulte en absoluto sorprendente que un significativo número de usuarios de ordenadores personales incluyan la posibilidad de mejorar las oportunidades educativas de sus hijos como una de las principales razones que los inducen a comprar una máquina.

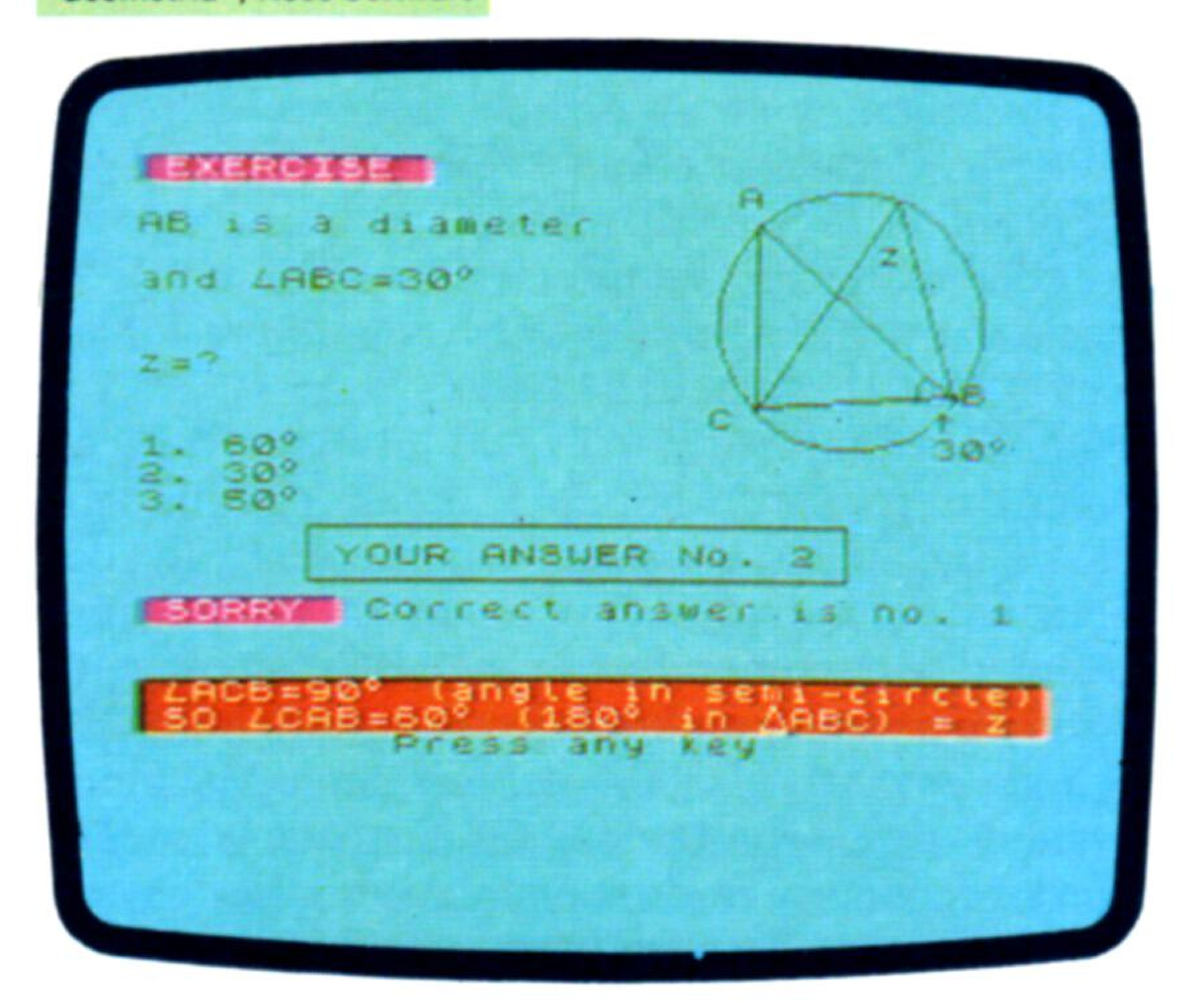
Pero dejando a un lado el problema del hardware, no cabe duda de que los programas educativos juegan un papel cada vez más importante como medios auxiliares de enseñanza en la clase. Escribir software de repaso, como se ha dado en llamarlo, es un asunto directo, que requiere pocos de los trucos y recursos que se emplean en los programas para juegos basados en pantalla, por ejemplo. Se les considera libros de texto interactivos porque tienen mucho en común con sus equivalentes impresos. El nombre del autor, por ejemplo, puede influir de manera preponderante en las ventas, al igual que ocurre con un libro de texto, y el patrón narrativo establecido en literatura se continúa, asimismo, en el nuevo medio.

La diferencia básica está, no obstante, en la introducción del concepto de interacción. Tradicionalmente, se espera que los estudiantes se basen en dos fuentes de conocimientos: el maestro y el libro de texto. La relación con el maestro es, hasta cierto punto, interactiva, aunque sólo en un pequeño grado, debido a que en cada clase suele haber 30 alumnos o más, y dos horas de instrucción por semana significan que, en el mejor de los casos, cada niño recibe cuatro minutos de atención exclusiva del maestro. No es sorprendente que los educadores hayan explorado las posibilidades de formas más eficaces de que los niños interactúen con el material que están aprendiendo.

Los primeros intentos de cara al aprendizaje mecanizado fueron los laboratorios de idiomas de la década de los sesenta, que se aplicaron a la enseñanza de temas imitativos como son los idiomas extranjeros. Cada estudiante tenía acceso a una grabadora y a un texto pregrabado, y trabajaba con este texto a su propio ritmo. El instructor tenía acceso al canal de audio individual de cada estudiante y podía supervisar o intervenir en la medida de lo necesario. Pero el objetivo consistía en estructurar el curso de instrucción de modo tal que sus intervenciones fueran lo más aisladas posible.

La enseñanza basada en ordenador lleva este proceso a su conclusión lógica y elimina la necesidad de un instructor que intervenga en un momento dado. En muchos sentidos, el paquete de repaso ha de ser un sistema especializado: es decir, tiene que ser totalmente autocontenido, debe evitar las imprecisiones y, además, es necesario que esté diseñado de modo que conduzca al usuario a través del material de una manera fácil y natural. En este caso el sistema informático debería ser lo más sencillo posible de utilizar. Debe partir del supuesto de que los conocimientos informáticos del usuario son mínimos y, por cierto, para ser más manejable no ha de requerir en absoluto de la pericia propia de un programador o un operador de ordenadores.

"Geometría", Rose Software



Cuando llega el momento de considerar el contenido de software, primero se debe diferenciar entre las ciencias (en las que tratamos con hechos conocidos y cuantificables) y las humanidades (en las que gran parte del análisis es de índole subjetiva y producto, por tanto, del criterio personal). En el caso de los paquetes de repaso basados en ordenador, hay aún otra distinción entre las humanidades y las ciencias. Estas últimas tenderán a estar muy ilustradas (aun cuando debamos depender de microordenadores que posean capacidades para gráficos de resolución relativamente bajas) mientras que las materias de humanidades se verán obligadas a basarse sobre todo en rutinas de manipulación de textos.

Además de los paquetes de repaso y de los paquetes muy especializados en un plan de estudios, en el campo de la educación básica se ha creado gran cantidad de material, en especial las operaciones aritméticas, lectura y ortografía.

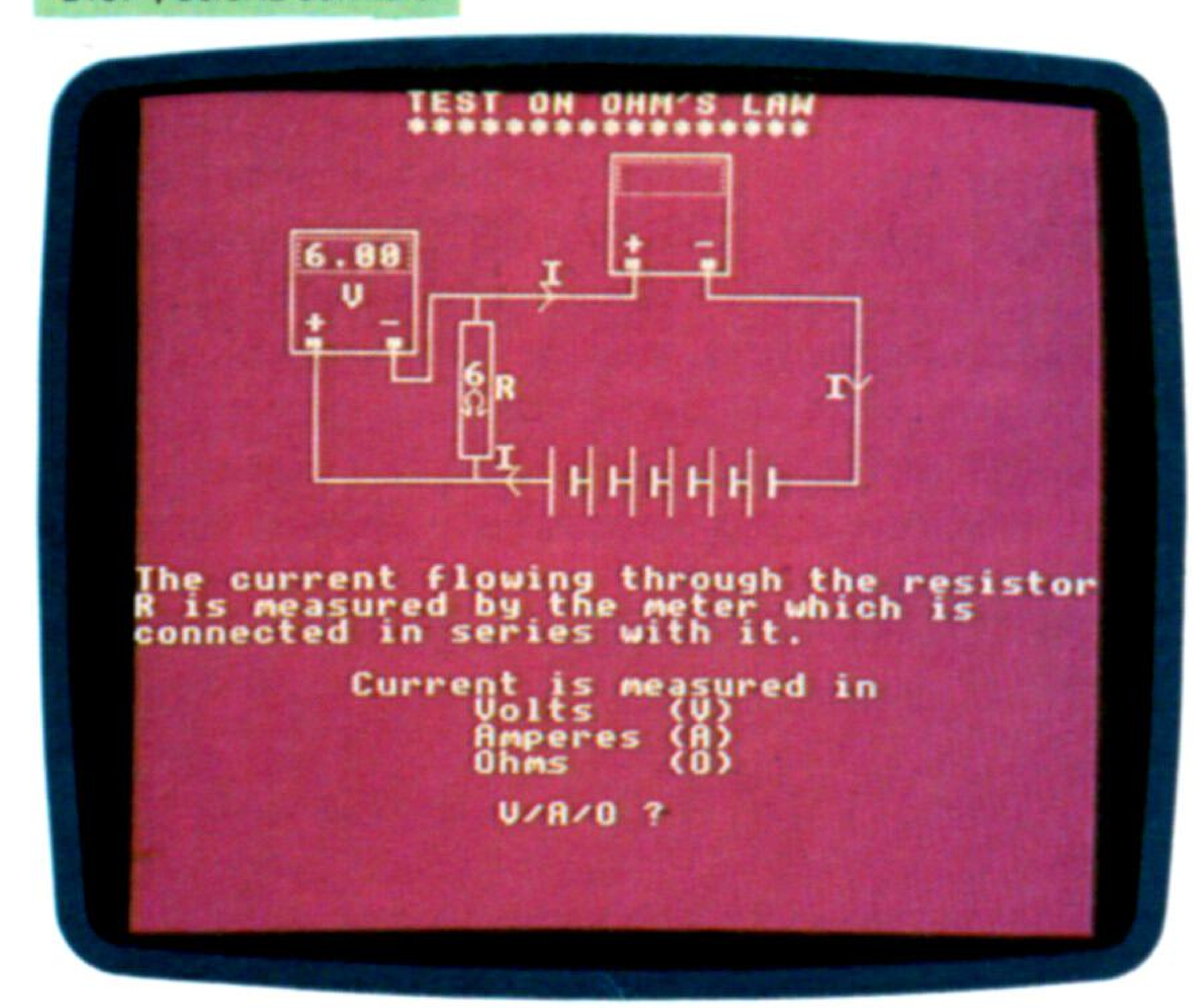
Hasta ahora nos hemos venido ciñendo al software de carácter didáctico o bien examinador, ya sea tratando de presentar los hechos de modo que resulten fáciles de memorizar, o bien formulando preguntas tipo test, cuya respuesta correcta está incluida entre otras opciones. Sin embargo, existe un tercer tipo de software, que por lo general el alumno utiliza más en la clase que en su casa: el software de simulación experimental. En este caso las técnicas de programación empleadas son mucho más complejas, ya que se requiere que el programa reproduzca matemáticamente la interacción de fuerzas físicas. La aplicación de métodos de simulación

en la clase está consiguiendo entre los profesores de ciencias tanta popularidad como la que obtuvo con sus equivalentes en la industria, por la sencilla razón de que permite en el ámbito de la clase un nivel de experimentación muchísimo mayor, en especial en aquellas áreas que, en virtud del costo o de los peligros inherentes a la utilización de químicos reactivos, de otra forma resultaría imposible su puesta en práctica.

Consideremos ahora los tres principales grupos por edades para los cuales se producen paquetes educativos, con el fin de comparar las similitudes y las diferencias del software.

Hasta los ocho años

La mayoría de los paquetes de software destinados a los niños más pequeños están relacionados con el desarrollo de la destreza básica en cuanto al reconocimiento de formas y patrones. Muchos utilizan la aritmética simple o la ortografía como los objetos a reconocer o emparejar, reforzando, por consi-



"D.C.", SciCAL Software

guiente, la familiaridad con estos símbolos. Muchos programas que entran en esta categoría se valen de "guiones" del tipo de los de los juegos con la intención de captar la atención del niño con mayor eficacia. Algunos presuponen la presencia de un adulto o de algún niño mayor.

En este grupo de edades gozan de especial aceptación los programas que le enseñan al niño a decir la hora, contar, sumar y restar (un método interesante se sirve de una báscula animada que se inclina hacia uno u otro lado según la carga de cada plato), construir oraciones cortas y deletrear palabras corrientes.

De nueve a catorce años

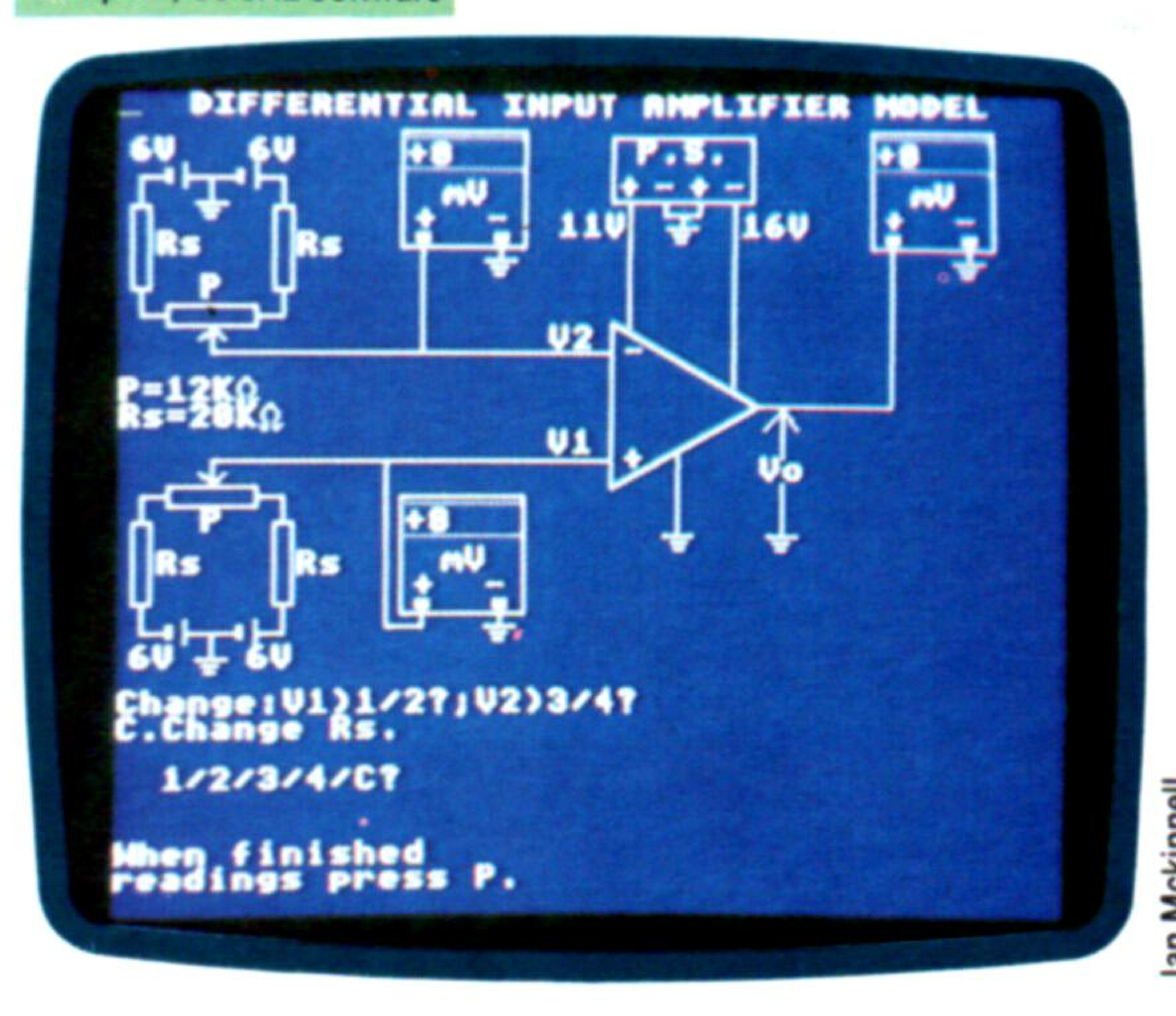
Para estas edades se abandona un poco el aprendizaje a través del juego y se adopta un enfoque algo más sofisticado, que, en este sentido, reproduce las experiencias del niño en la clase. No es sorprendente que, dado que se espera que el software de este tipo logre que el niño se sienta motivado a utilizarlo, se hayan dedicado considerables esfuerzos en cuanto a incentivos y recompensas. Un método que se ha hecho popular y ha obtenido un éxito razonable le presenta al niño un juego en pantalla después de que él (o ella) ha completado una sección del programa de aprendizaje dentro del tiempo permitido y con un nivel dado de éxito.

Para los niños entre estas edades los temas más populares siguen siendo la aritmética, la ortografía y el empleo del lenguaje. Pero, además, se encuentran a la venta diversos paquetes de temas históricos o geográficos, algunos que enseñan la teoría básica de la música y cierto número de simulacros sencillos.

A partir de los quince

Es en esta etapa cuando hacen su aparición los paquetes de software educativo de tipo sistema especializado. Existen, por supuesto, paquetes para reforzar las destrezas básicas, pero la mayoría de los fabricantes concentran todos sus esfuerzos en áreas de temas particulares. En este nivel los programas individuales suelen ofrecer un resumen de los conocimientos básicos, proporcionan la teoría y la metodología del tema en cuestión e incluyen una larga relación de preguntas tipo test, con respuestas múltiples, basadas en la asignatura de estudio. En materias como literatura, también ofrecen un análisis de estilo y contenido, de la misma manera que lo haría un profesor.

Se espera que, llegado a esta edad, el alumno haya alcanzado un cierto nivel de automotivación, y se dedica poco esfuerzo, o ninguno, por mantenerlo interesado en el trabajo que tiene entre manos mediante medios artificiales. Esto no quiere



"Ampli.", SciCAL Software

Ian Mckinnell

decir que el tema central se suela presentar de manera convencional. Por el contrario, se propicia universalmente un enfoque inventivo de los gráficos estáticos y animados y el empleo del sonido sintetizado. Para los estudiantes mayores de quince años existe una variedad de paquetes software disponibles, y como muchos de ellos están orientados a un tema muy específico, lo mejor siempre será consultar a un profesional especializado en software educativo para formarse una idea cabal acerca de sus valores relativos.

En un próximo artículo analizaremos con mayor detalle la gama de paquetes de software educativo disponibles en el mercado para los ordenadores personales más populares.

Funciones y estructuras de control

Examinadas las variables y las cadenas en el BASIC del Spectrum, nos falta analizar algunas funciones de este lenguaje, como VAL, WHILE...WEND y REPEAT...UNTIL

Puede que ya haya observado que en el BASIC de Sinclair algunas funciones no requieren encerrar sus argumentos entre paréntesis, a diferencia de sus equivalentes en otras versiones de BASIC, de modo que LEN (X$) se puede escribir ya sea como LEN X$ o LEN(X$). No obstante, es obligado utilizar los paréntesis cuando el significado de una expresión sea dudoso o ambiguo.

La función CODE es el equivalente Sinclair de ASC() y se comporta exactamente de la misma manera. Sin embargo, el juego de caracteres Sinclair es ASCII estándar sólo para los valores entre 32 y 122. De modo que, por ejemplo, si en la mayoría de las versiones PRINT CHR$ (7) da un sonido, el Sinclair envía un mensaje de error.

La función VAL es del BASIC común, pero en Sinclair una sentencia con VAL("a45") rompería el programa, porque el argumento de la función es no numérico. En la mayoría de las otras versiones esto sencillamente daría el valor cero. Si tal arbitrariedad constituyera para usted un problema, escriba una subrutina para sustituir la función VAL, o verifique el valor CODE del primer carácter del argumento de VAL: si CODE A$ (1) < 48 o CODE A$(1) > 57 entonces A$ es no numérica y no puede ser el argumento de la función VAL.

La VAL de Sinclair, no obstante, puede evaluar expresiones numéricas; de modo que:

```
LET A$ = 6*12":PRINT VAL A$
```

daría 72, el valor de la expresión "6*12". En la mayoría de los BASIC la función VAL no es tan poderosa y, en este caso, daría el valor 6.

Esta capacidad de evaluar expresiones se puede utilizar de muchas maneras. El siguiente programa para dibujar un diagrama de barras es un sencillo ejemplo:

```
100 DIM S$(31)
200 LET S$ = "*****************************
    **"
300 INPUT "DE ENTRADA A UNA FUNCION DE X";F$
400 PRINT "Y = ";F$:PRINT
500 FOR X = 1 TO 10
600 PRINT S$(TO INT(VAL F$))
700 NEXT X
800 PRINT "==================
    =========="
900 PRINT "0000000001111111111222222222223"
950 PRINT "1234567890123456789012345678890"
```

Cuando ejecute este programa, podría digitar una expresión algebraica con variable X como la siguiente: 2*X + 3/X y vería en la pantalla un diagrama de barras de dicha función. En este programa el eje *y* es horizontal, el eje *x* vertical y los gráficos son del tamaño de un pixel. No se requerirían muchas más instrucciones para dar una escala a los valores de *x* e *y*, inmediatamente después de dar entrada a la función de *x*, y luego hacer ajustes de escala, imprimir ejes y crear gráficos en alta resolución. En resultado sería un paquete gráfico muy llamativo; sólo que el fragmento anterior se ha diseñado simplemente para clarificar aún más la gran utilidad del poder de VAL para aceptar expresiones de argumento.

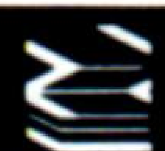
En este sentido, GOSUB y GOTO son iguales que VAL. También aquí se admiten expresiones cuando la mayoría de las versiones de BASIC exigen que sus argumentos sean números de línea concretos. Se consiguen varias ventajas. Puede, por ejemplo, ponerles nombre a sus subrutinas, definir variables con los mismos nombres y valores apropiados, y después ir hacia las subrutinas (GOSUB) por su nombre. He aquí un ejemplo de esta capacidad:

```
100 LET INIC = 1000
200 LET RESULT = 2000
300 LET CALCULO = 3000
400 GOSUB INIC
500 GOSUB CALCULO
600 GOSUB RESULT
700 STOP
1000 REM******* S/R INIC********
............
2000 REM***** S/R RESULT********
............
3000 REM**** S/R CALCULO******
```

que casi hace que su programa se documente a sí mismo. Si reemplazara GOSUB INIC por GOSUB (VAR1 + N*VAR2), o por cualquier expresión numérica válida, se evaluaría la expresión y se trataría al resultado como un número de línea. Además, en Sinclair, si el argumento de GOTO o GOSUB es un número de línea que no existe, entonces el control pasa al siguiente número de línea válido. Si encuentra, por ejemplo, GOTO 17, y la línea 17 no existe pero sí existe la línea 18, entonces el control pasará a la línea 18 y no se indicará ningún error, como ocurriría en la mayoría de las versiones de BASIC.

La capacidad del BASIC de Sinclair para manipular estos "saltos calculados" compensa la carencia de estructuras ON...GOTO y ON...GOSUB. En otro BASIC se podría escribir:

```
2360 ON D GOSUB 100,200,300,400,500
```



para dar a entender que si el valor de D es 1, entonces el programa saltará a la línea 100 (GOSUB 100), si D = 2 entonces saltará a la línea 200 (GOSUB 200) y así sucesivamente. En el BASIC de Sinclair estaría capacitado para escribir:

```
2360 GOSUB (100*D)
```

para producir exactamente el mismo resultado. Sin embargo, en ambas versiones es imprescindible prever lo que sucedería si el valor de D no fuera entero, si fuera menor que uno, o mayor que cinco.

El Spectrum no admite las estructuras de control REPEAT...UNTIL ni WHILE...WEND. Éstas, no obstante, se pueden simular de diversas formas. La estructura REPEAT (repetir) es un bucle que empieza en la palabra REPEAT y termina en la palabra UNTIL (hasta), que es seguida por una sentencia condicionada. Si esta condición es verdadera, entonces el bucle se acaba y el control pasa a la sentencia que sigue a UNTIL. En el Spectrum, esta estructura de control se puede imitar de esta manera:

```
100 DATA "A","B","C","D","E","*","F","G","*"
200 FOR L = 1 TO 1
300 READ X$
400 PRINT X$
500 IF X$ <> "*" THEN LET L = 0
550 NEXT L
600 PRINT"FIN DE DATOS"
```

El problema de la estructura REPEAT...UNTIL es que, debido a que la condición de salida se verifica al final del bucle, éste siempre se ejecutará al menos una vez. Si no se desea que resulte de esta forma, entonces se debe utilizar la estructura WHILE...WEND.

Esta estructura es un bucle que empieza en la sentencia WHILE (mientras), a la que sigue una condición y que termina en la sentencia W(HILE)END (fin de WHILE). En la medida en que la condición se cumpla, las sentencias del programa entre WHILE y WEND se ejecutarán repetidamente. Cuando la condición no se cumpla, el control pasa de la sentencia WHILE a la sentencia que sigue a WEND (saltándose las sentencias entre ambas). El programa que dimos, por consiguiente, sería así:

```
100 DATA "A","B","C","D","E","*","F","G","*"
200 WHILE X$ <> "*" DO
300 READ X$
400 PRINT X$
500 WEND
600 PRINT"FIN DE DATOS"
```

En las máquinas Sinclair esto se podría sustituir por una estructura IF...THEN...GOTO, pero otra alternativa es:

```
50 LET X$ = ""
100 DATA "A","B","C","D","E","*","F","G","*"
200 LET TEST = (X$ <> "*")
250 FOR L = 1 TO TEST
300 READ X$
400 PRINT X$
500 LET L = (X$ = "*")
550 NEXT L
600 PRINT "FIN DE DATOS"
```

Hemos examinado las principales variantes del BASIC de Sinclair. Existen otras pequeñas diferencias, algunas de las cuales son trabas más que genuinas modificaciones, pero con cuidado y cón el manual podrá hacerles frente.



Mecanografía visual

El tratamiento de textos es la aplicación reina de la microinformática profesional; paquetes de software como el WordStar están entre los más vendidos en todo el mundo

En su forma más simple, un procesador de textos no es más que una perfecta máquina de escribir. La pantalla del monitor sustituye al papel, y la tecla DELETE actúa como una goma de borrar. Para el usuario de gestión, lo más deseable de un sistema para tratamiento de textos en su capacidad para guardar documentos (en disco o en cinta de cassette), permitiéndole luego que esos documentos (que pueden ser líneas individuales, oraciones, párrafos o páginas enteras) se incorporen a un texto recién dictado. Esto es de particular utilidad en la producción de especificaciones, contratos u otra clase de documentos que emplean el mismo texto una y otra vez.

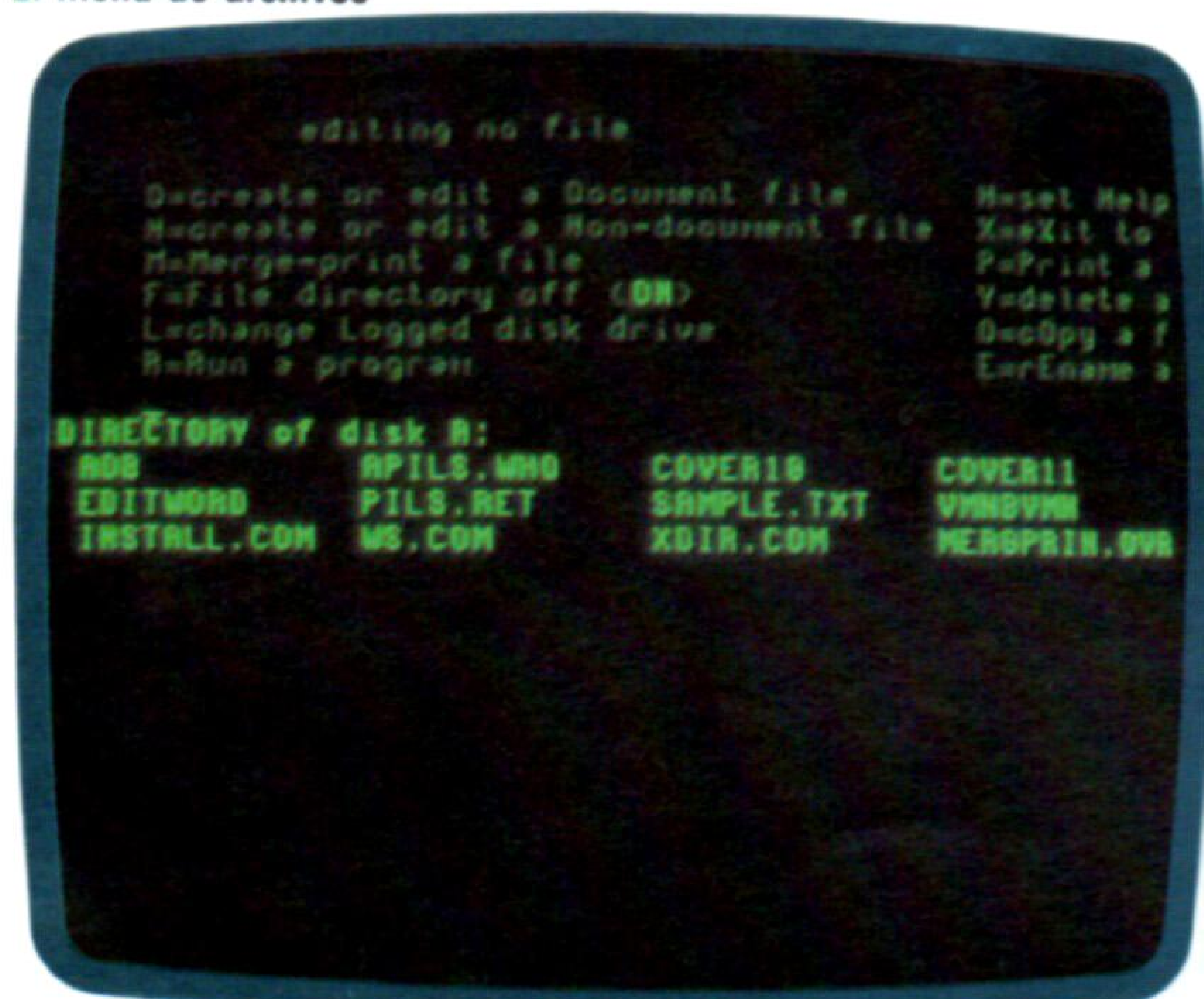
Cuando ya se conoce cómo crear, modificar y manipular un documento, almacenarlo y recuperarlo, el siguiente aspecto a tratar es cómo reacomodar el texto en la página. Existen tres formas convencionales de disponer el texto: alineado sobre la izquierda (cada línea deja el mismo margen izquierdo), alineado sobre la derecha, o justificado. En el texto impreso justificado, los espacios entre palabras y letras están dispuestos de tal modo que cada una de las líneas poseen la misma longitud. Sin embargo, la mayoría de las impresoras para microordenadores que hay en el mercado sólo varían los espacios en blanco entre palabras cuando componen texto justificado.

Igualmente un paquete para tratamiento de textos debería ser capaz de buscar a través de todo el documento una palabra o una frase determinada y sustituirla por otra. Esto es particularmente útil cuando se ha escrito de manera reiterada con algún error de ortografía una misma palabra, o cuando se desea enviar correspondencia personalizada a alguien determinado. En términos técnicos, WordStar es un procesador y editor de textos activado por menú, con órdenes de formato transparentes, que funciona en un medio CP/M y requiere la utilización de la tecla Control para permitir que la máquina distinga entre una orden y un carácter. Ello significa que el operador puede elegir tener a su disposición un menú de opciones que esté visualizado constantemente en la pantalla, en este caso en una sección reservada de diez líneas en la parte superior. Las órdenes que afectan a la forma en que aparece el texto (ya sea en el momento de entrada o durante la impresión) se visualizan en la pantalla dentro del cuerpo principal, pero la impresora no las reproduce; la responsabilidad de la manipulación de archivos y "administración" la asume el sistema operativo.

Una vez cargado, WordStar presenta al usuario un "menú" que permite seleccionar la función requerida. El usuario puede crear o editar un archivo, ya sea como documento o como "no documento" (que se utiliza, p. ej., en los programas en lenguaje ensamblador que se van a compilar, o para el texto que se vaya a utilizar con otro paquete de software); puede también mezclar archivos entre sí (imprimiéndolos o no); cambiar la unidad de disco en uso de A a B, o viceversa; eliminar, copiar o darle un nombre nuevo a un archivo; salir al sistema operativo, o establecer el nivel de ayuda requerido.

Cada opción se selecciona mediante una sola pulsación de tecla (no se necesita la tecla Control), lo que conduce al usuario hasta un menú más pequeño, específico para la función en uso. La última de estas opciones (para determinar el nivel de ayuda) le permite al usuario seleccionar en qué medida se visualizará en la pantalla el menú de órdenes disponibles. Cuanto más bajo sea el nivel de ayuda requerido, menos serán las líneas reservadas en la

El menú de archivos



El menú de ayuda



El menú de archivos
Para empezar, WordStar presenta un menú que revela la gama de opciones de sus funciones básicas. El usuario puede crear o editar ya sea un archivo de documentos utilizando todos los parámetros de formateo estándar, como traslado de palabras, justificación, etc., o un archivo de no documentos, o copiar, dar un nombre nuevo o borrar un archivo. En resumen, esta sección del programa se ocupa de las rutinas de administración. Estas fotografías corresponden al Osborne-1, que visualiza sólo 52 columnas de texto de las 80 existentes, cortando por consiguiente sobre el lado derecho

El menú de ayuda
Este menú, al que se accede mediante el código de Control (J), le ofrece al usuario una serie de explicaciones en pantalla bastante comprensibles acerca de la forma en que funcionan ciertas secciones del paquete. Coloquialmente se lo conoce como menú de ayuda. El código de control J seguido de uno de los caracteres subsidiarios (H, B, F, D, etc.) ofrece una explicación más extensa de estas secciones del WordStar



pantalla para el menú, y más líneas quedarán reservadas para la edición de texto. La mayoría de las órdenes de tecla con iniciales son siglas, pues se tratan de un único carácter. Por ejemplo, D crea un archivo para "documento" y N crea un archivo para "no documento".

Lo primero que solicita el sistema es un nombre para el archivo que se va a crear. Una vez dado un nombre al archivo, el sistema pasa al menú de trabajo, que consta de indicaciones para detalles como movimiento del cursor, inserción y eliminación, y recordatorios de los códigos de control de un único carácter utilizados para dar entrada a otras bifurcaciones del menú principal (el menú de formato, p. ej.). El WordStar no da por sentado que la máquina en la cual se va a ejecutar (el ordenador "objeto") posee teclas para control del cursor.

El movimiento de un solo espacio se define mediante la tecla Control en conjunción con las teclas E, S, D y X (las máquinas que poseen teclas para control del cursor normalmente tienen instalado el WordStar, de modo que las funciones se duplican). El WordStar es bastante eficaz en su control del cursor, acomodando el movimiento de éste por carácter, palabra, línea o párrafo en cualquier dirección. Para la supresión rige casi el mismo principio. Los caracteres individuales se pueden borrar tanto en la posición del cursor como en la posición inmediatamente a su derecha; y se pueden suprimir palabras enteras a la derecha de la posición del cursor o líneas enteras (independientemente de la posición del cursor). También se puede "enrollar" la pantalla utilizando las funciones de control, sin afectar el posicionamiento del cursor. Esto es muy útil cuando se quiere revisar el texto.

Por último, el usuario puede decidir si insertar un carácter en la posición del cursor o reemplazar el que está escrito por otro que desea introducir desde el teclado. Esta función se realiza mediante la presión de la tecla apropiada (V, en este caso), que cambia el estado de inserción a sustitución, y así permanece hasta que se vuelve a pulsar la tecla V (junto con la tecla Control), para invertirlo.

El paquete se encarga automáticamente de la paginación y del traslado de palabras al final de las líneas. La paginación define el número de líneas por página y el traslado de palabras es el desplazamiento automático de palabras a la línea siguiente.

Todas las funciones que hemos descrito son las necesarias normalmente para crear y editar documentos, pero el WordStar posee otras configuraciones. Puede centrar líneas en la página; posibilitar golpeteos múltiples para obtener caracteres en negrita; poner caracteres sobrescritos y subscritos; sobreimprimir la línea anterior, o subrayar y tachar caracteres individuales. Todas estas funciones son utilizables en el momento de dar entrada al texto o durante la edición.

A la función "buscar y sustituir" se accede a través del menú Q, que también contiene el posicionamiento extra del cursor y las funciones de borrado. El usuario recibe aviso de que dé entrada a la serie de caracteres que componen la palabra o la frase a buscar. El usuario puede mencionar una instancia o bien todas las instancias, y decidir si la serie localizada sencillamente la debe señalar el cursor, o si se ha de sustituir y, de ser de esta forma, si esa sustitución ha de efectuarse sin que medie ninguna otra instrucción.

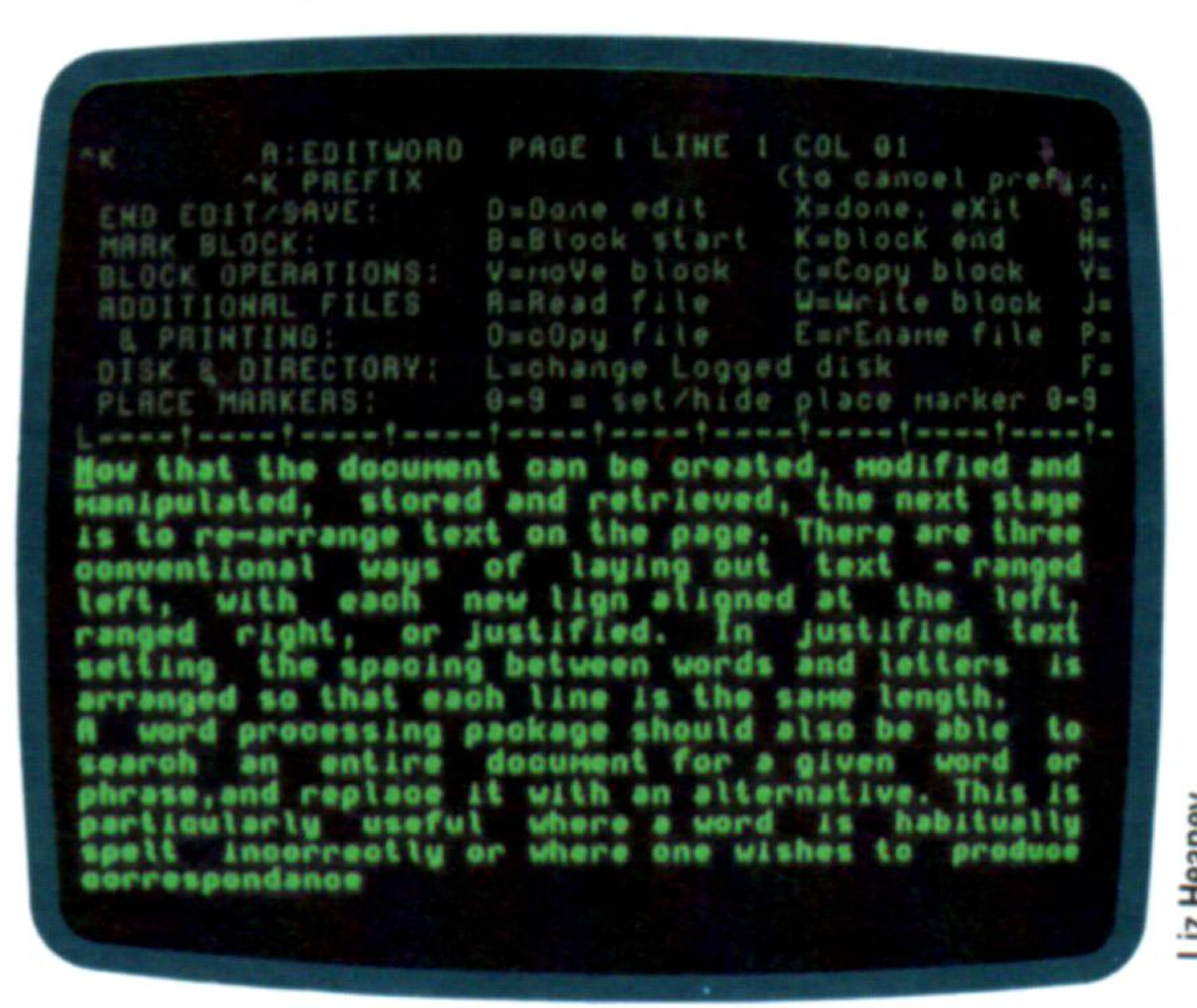
Igualmente, se pueden desplazar de un lugar del documento a otro bloques marcados de texto mediante diversas funciones del menú K, que de otro modo se ocupa de manipular el archivo. J, O y P, los otros tres menús subsidiarios del WordStar, se encargan, respectivamente, de ayudar al usuario a crear o editar un documento, del formateo y ajuste de los documentos, y de la forma en que aparecerán al imprimirse.

En líneas generales, el WordStar es un paquete de software para microordenadores profesional y amplio. Son muy pocas las funciones que no puede realizar satisfactoriamente, y está definido de tal forma que resulta accesible para una amplia variedad de máquinas. A primera vista podría parecer complicado y diferente, pero gracias a sus exhaustivos menús hasta los principiantes pueden utilizarlo.

Un importante número de los microordenadores que admiten el WordStar poseen teclas de función programable como estándar. El Osborne-1, por ejemplo, con el cual hemos visualizado nuestros ejemplos, le ofrece al usuario 10 de estas teclas y reserva 256 bytes para sus contenidos. Estas funciones (a las que se accede mediante la tecla Control y una tecla numérica que se pulsa simultáneamente) pueden contener una serie de caracteres de control, o una palabra o frase que se emplee comúnmente, y posibilitar que esa serie de caracteres se incorpore mediante apenas dos pulsaciones de teclas.

El menú de órdenes rápidas

El menú de bloques



Liz Heaney

El menú de órdenes rápidas
A los menús secundarios del WordStar se accede mediante un código de control de un solo carácter, que hasta cierto punto es acrónimo. El menú Q, por ejemplo, equivale a decir en forma abreviada el menú Quick (rápido). El menú Q proporciona un resumen de las teclas de control del cursor, así como las funciones de buscar y sustituir del paquete

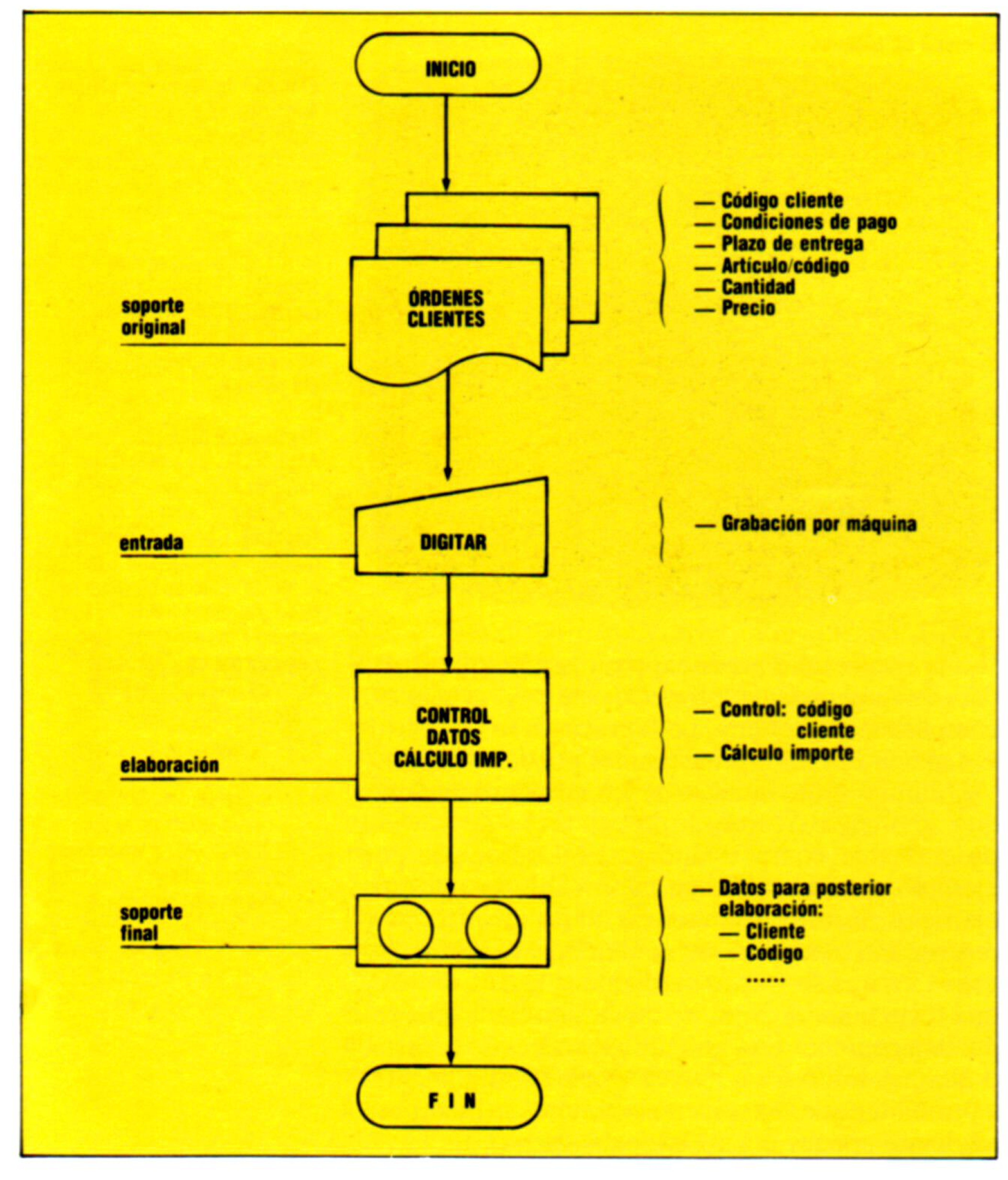
El menú de bloques
Algunas de las siglas de un solo carácter son algo arbitrarias; K, por ejemplo, alude a *blocK* (bloque). Controla cómo se guardan y utilizan los archivos, el movimiento de bloques de texto y algunas estructuras de acceso a los archivos. Observe que reproduce, asimismo, algunas de las rutinas de administración estándar del paquete

Diagramas de flujo

Los diagramas de flujo constituyen un punto de referencia para el analista que acaba de concluir un primer estudio global del problema

Ya hemos presentado los diagramas como la secuencia lógica de operaciones a seguir que forman un programa. Uno de los propósitos confiados al análisis es planificar la combinación de los diferentes componentes del sistema, para alcanzar, en el menor tiempo posible, los mejores resultados. Debemos analizar el modo cómo fluyen los datos a través de los citados componentes, y, tras individualizar esta corriente de datos por medio del sistema, se generaliza la secuencia de las operaciones necesarias para poder codificar los símbolos. Está claro que esta parte del análisis no permite por sí sola confeccionar un programa.

La diagramación, aplicada a la automatización de la información, es en sí, como término, demasiado genérica. En cuanto a los diagramas, empecemos distinguiendo dos tipos básicos: *diagrama de flujo*, llamado también *flow-chart* de sistema, y *diagrama de bloques*, o *flow-chart* de programa. Aun tratándose de la resolución de un mismo problema, en realidad cada uno cumple una función específica, puesto que ambos responden a diferentes exigencias, aunque complementarias entre sí.



Un diagrama de flujo busca la representación gráfica del paso de los datos, explicando, a grandes rasgos, la resolución y su proceso. Define igualmente la configuración necesaria para resolver el problema, mencionando sintéticamente las operaciones, pero sin detallar los movimientos de cada frase, con lo que, a cambio de una visión global, nos hallamos faltos de información específica para poder transcribir y crear el programa.

Pero ¿para qué sirve un diagrama de flujo? Podemos resumirlo en estos puntos: describe todas las fases de un proceso; asigna y define aquellos componentes del sistema que se utilizarán en cada una de las fases; relaciona las fases y sus componentes.

Supongamos que una pequeña empresa tiene la intención de automatizar su sistema de pedidos y sustituir sus archivos tradicionales por uno informático. La persona encargada de esta tarea (el analista) recibe una serie de explicaciones acerca de la manera en que se había desarrollado el trabajo hasta la fecha: "Los clientes hacen los pedidos que nosotros recogemos en unos albaranes. Éstos contienen datos tales como número de código, nombre del cliente, artículo, número de unidades, etc. Buscamos después en los archivos la ficha del cliente y una vez cotejados los datos, se le aplican las condiciones de pago, entrega y otros extremos que en ella figuren. Enseguida calculamos el importe del pedido. Finalmente archivamos, corregida, la ficha".

Nuestro analista, que bien pudiera ser usted, dando por comprendido el método expuesto y habiéndose cerciorado de que la empresa no desea introducir variaciones de ningún tipo, estudió el problema llegando al siguiente planteamiento global. Según éste, los pasos esenciales a desarrollar han de ser:

Recibir el soporte original formado por las órdenes de pedido

Introducir los datos en el ordenador

Realizar las pertinentes comprobaciones y una vez hechos los controles, calcular a cuánto asciende el total del pedido. Este paso se realizará íntegramente de forma automática

Proceder a la grabación del registro correspondiente al cliente, debidamente actualizado en el soporte (magnético) final

Éste fue el planteamiento que pudo entregar usted a la figurada empresa con el diagrama que aparece aquí al lado.



Una buena imagen

La superior calidad de imagen que proporciona un monitor respecto a un televisor puede ser decisiva para un ordenador que se desea emplear en gestión empresarial o en la enseñanza

Entre los periféricos más importantes se encuentra la pantalla. Es frecuente el uso de un televisor o un aparato de bajo coste en blanco y negro adquirido especialmente pero, aunque la calidad obtenida sea aceptable, no es todo lo buena que debiera ser.

Esto se debe a que la señal, al pasar de la memoria de pantalla del interior del ordenador a la pantalla, debe someterse a varias etapas de codificación y decodificación. Con independencia de la bondad del sistema de circuitos usado, inevitablemente será poco perfecto, con una imagen final a menudo borrosa y difícil de leer, afectada por un desagradable rehílo lumínico, que se conoce como "hormigueo de puntos".

El secreto en una visualización de calidad está, pues, en eliminar estas distorsiones de señales; y dado que el elemento que las produce es el sistema de circuitos de modulación y demodulación, se trata sencillamente de no incluir dichos sistemas en el proceso.

En realidad esta tarea es la que realiza un monitor, que no consiste más que en un tubo catódico sin los decodificadores de televisión, un sistema simplificado que tiene capacidad para generar imágenes más nítidas, más brillantes y, por lo general, más estables.

Puesto que no lleva decodificadores de televisión, el monitor no funcionará si estuviera conectado a la salida TV de su ordenador. Necesita una salida "video". Puede que en su ordenador no se llame de este modo, pero de todas formas el detalle más importante es que debe tratarse de una salida que no pase a través del modulador y para comprobar esta característica usted debe consultar el manual de su ordenador.

El proceso que genera la imagen en una pantalla de televisión consiste en asegurarse de que todo suceda en el momento correcto. El problema radica en el hecho de que el barrido del haz a través de la superficie del tubo se genera dentro del monitor y, por lo tanto, es inaccesible para el dispositivo activador del propio ordenador.

Estando encendido pero sin conectar a una entrada, un monitor explorará el haz sobre la pantalla entera 50 veces por segundo, produciendo un campo perfectamente iluminado. Convertir éste en una imagen implica encender y apagar este haz exactamente en el mismo lugar cada 1/50 de segundo; cualquier inestabilidad que se produzca durante este proceso se traducirá en un cabrilleo molesto, que hace que mirar la pantalla resulte, en el mejor de los casos, fatigoso y, en el peor, absolutamente inútil.

Todo el proceso depende de *impulsos de sincronización* que, junto con la señal de brillo, son producidos directamente por el ordenador y salen hacia el monitor.

Hay dos tipos de impulsos de sincronización: uno para cada línea de la imagen y otro para cada imagen completa. Al final de cada ciclo completo se le envía al monitor un impulso corto que le indica que ahora el fotograma está completo y que el haz de electrones (y, por tanto, el punto que produce) ha de regresar a la esquina superior izquierda del fotograma para repetir el ciclo.

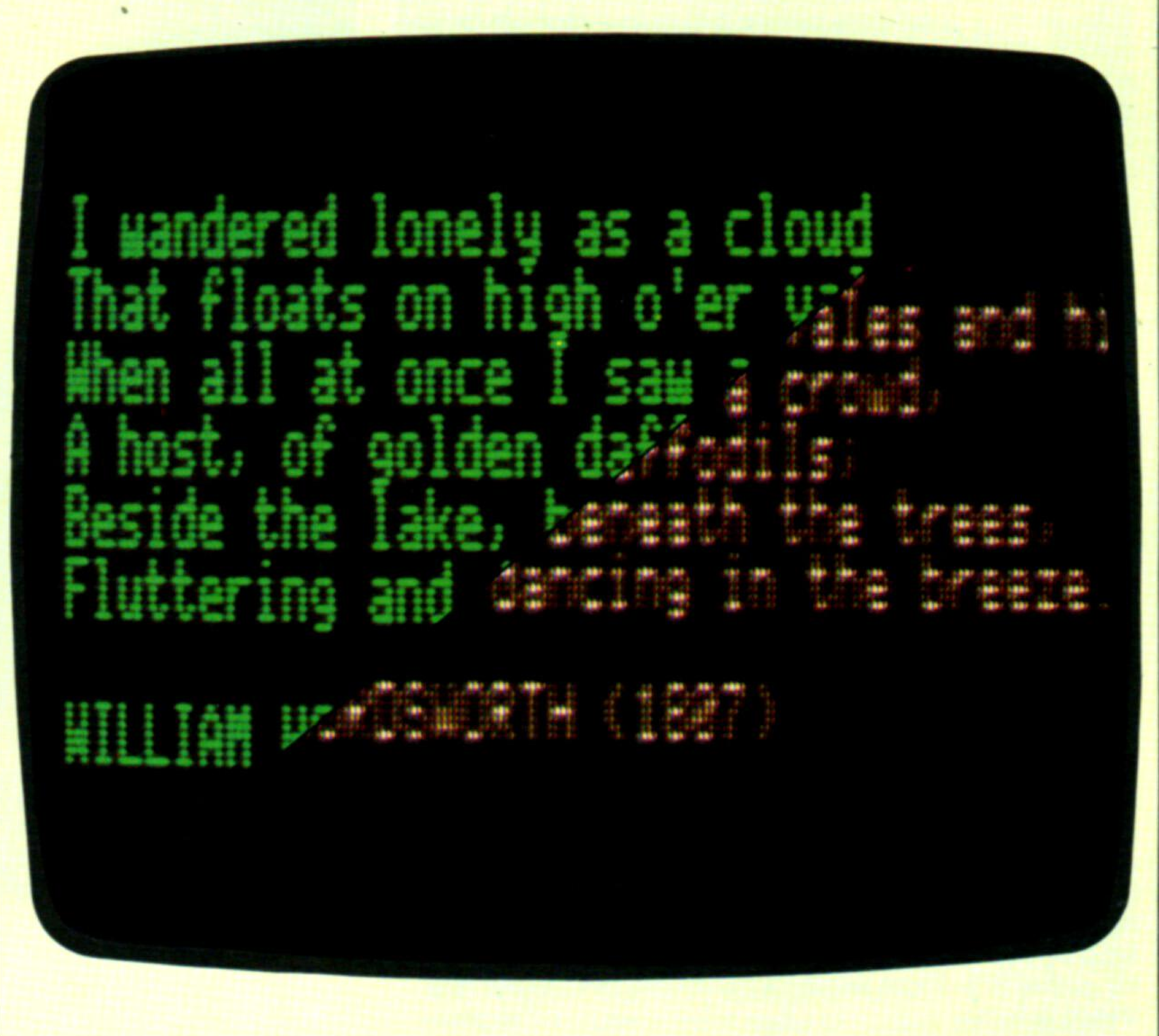
Algo similar se produce al final de cada línea. El monitor recibe un impulso que le indica que una línea está completa y que el haz de electrones debe regresar al lado izquierdo de la pantalla para dar comienzo a la línea siguiente.

Hay muchos modelos de monitores, y sólo dos clases: en color y monocromos, que a su vez se dividen según los diferentes tipos de señales que aceptan.

Los monitores monocromáticos son bastante sencillos y el único tipo importante es el que acepta

Especialista en imágenes

Esta pantalla ilustra la diferencia entre la calidad de imagen de un monitor monocromático (en este caso el Monitor III de Apple) y un buen televisor doméstico utilizado para visualizar la salida de un paquete para tratamiento de textos



Ian Mckinnell

una señal denominada *compuesta*, así llamada porque los diversos impulsos de sincronización se combinan con el nivel de brillo en una señal, que el monitor clasifica después para la producción de la imagen.

Monitores en color

Algunos monitores en color utilizan un esquema similar, pero en virtud de la mayor complejidad de la señal están más emparentados con los aparatos de televisión y operan casi con los mismos sistemas. Los tipos principales son PAL, SECAM y NTSC, nombres de los sistemas de color que se utilizan en las emisoras de televisión de diferentes partes del mundo. Representan métodos de codificar los tres colores "aditivos" (rojo, verde y azul) y los dos impulsos de sincronización.

En algunos casos, las diversas señales se pueden enviar al monitor por separado. A pesar de que en los esquemas utilizados existen sutiles variaciones, genéricamente se conocen como RGB por las pistolas de color (*Red, Green, Blue*: rojo, verde, azul). El más simple es el TTL (*Transistor-Transistor logic*: lógica transistor-transistor), en el cual los colores sólo poseen dos estados, encendido o apagado, de modo que en diversas combinaciones pueden producir los ocho colores familiares que se aprecian en los sistemas de videotext como el Prestel y el Ceefax.

Es posible obtener más colores si se dota de intensidad a cada color aditivo; aunque tal intensidad suele producirse por pasos discretos (digitales), el

Anchura de banda

La anchura de banda es un extremo importante a la hora de elegir un monitor. Mide la pequeñez con que el sistema de circuitos puede producir un punto. Según esto, cuanto más pequeño sea el punto, mejor será; pero la conclusión sobre qué anchura de banda es mejor para una máquina determinada se basa también en otras consideraciones.

Antes de nada, usted debe calcular cuál es la anchura de banda mínima absoluta para su máquina. Ésta se calcula multiplicando el máximo número posible de líneas de caracteres en pantalla por el número de los caracteres de cada línea, lo que da el número total de caracteres. Esta cifra se multiplica, a su vez, por la anchura en puntos de la matriz de caracteres, y después por la profundidad de la matriz.

El resultado obtenido es el número máximo de puntos en pantalla, que por lo general se sitúa entre los 10 000 y 1 500 000. En una pantalla normal de 80 × 24 con una matriz de 7 × 9, la cifra es de 110 960. Todos estos puntos los ilumina el barrido en la pantalla a la velocidad de 50 veces por segundo (60 en Estados Unidos y otros países), lo que significa que el número de puntos de la pantalla se debe multiplicar por esta cifra. El resultado generalmente estará comprendido entre 500 000 y 75 000 000, que es el número de veces que el sistema de circuito de control debe poder encender y apagar el haz por segundo.

Como se trata de una frecuencia se expresa en hertzios (Hz) y, al igual que casi todas las magnitudes medidas por millones, las unidades son megahertzios o megas, para abreviarlo.

Los monitores se venden con anchuras de banda de 5, 7, 10, 12 y 15 MHz e incluso es posible que de 20 MHz, si bien el precio aumenta notablemente a medida que la cifra se eleva. Se pueden construir monitores que operen a 100 MHz, pero su precio sería excesivo.

Por lo general, a mayor anchura de banda del monitor, más nítida será la imagen, si bien una resolución excesivamente alta puede llegar a ser tan incómoda como una demasiado baja, en especial para los gráficos en color.

Si la anchura de banda del monitor es muy pequeña para el ordenador, las letras serán desdibujadas y difíciles de leer, como si la señal se enturbiara. Si, por el contrario, es demasiado grande, los puntos individuales se separarán con excesiva claridad, dividiendo la imagen.

Monitores monocromáticos
De todos los monitores, los má sencillos y menos caros son los modelos monocromos. A un precio razonable, estas unidade le ofrecen al usuario una opción de colores de fósforo y una definición razonablemente alta. Son ideales para tratamiento de textos y otras aplicaciones de gestión, pero también se utilizar para visualizar gráficos sencillos

Monitores a tres colores
El CUB de Microvitec es un excelente ejemplo de los monitores en color menos caros que están al alcance del usuario de un ordenador personal. Si bien su resolución no es de las más altas, es bastante adecuado para diversas aplicaciones, incluyendo la mayoría de los gráficos

Versatilidad japonesa
Con su capacidad para diferenciar la naturaleza de la señal recibida, el TM90PSN de JVC es especialmente valioso para quienes desean que su monitor tenga aplicaciones diferentes. Además de su empleo como dispositivo de salida de ordenador, también puede servir como aparato de video o reproductor de videodisco

INPUT SELECT 1 2



resultado se conoce como monitor *analógico* o *lineal*.

Los monitores en blanco y negro de nueve pulgadas suelen salir por la mitad del precio de uno con pantalla verde fósforo de 12 pulgadas. Los monitores en color son más caros, debido al mayor coste del tubo. Los monitores estándar, como el Microvitec y el Kaga, doblan el precio a los anteriores, dependiendo en gran medida de las dimensiones de la pantalla y de la anchura de banda. La mayoría de los monitores en color se pueden hallar tanto en versión TTL como analógica, con poca diferencia en cuanto se refiere al precio.

El TM90PSN es un monitor interesante y flexible que produce la firma JVC. Es más pequeño que la mayoría, con una pantalla de 10 pulgadas y, aunque la anchura de banda no es particularmente alta (lo cual limita la máxima resolución), puede aceptar casi cualquier tipo de entrada, desde la compuesta monocromática normal, pasando por la TTL y la RGB analógica, hasta cualquiera de las cuatro señales a color compuestas, PAL, SECAM, NTSC 3,58 y NTSC 4,43. Posee en su interior un mecanismo selector que verifica la señal entrante y automáticamente conmuta el monitor a la modalidad adecuada.

Este tipo de monitor se irá haciendo cada vez más común, y puede servir como salida con fines generales para ordenadores, aparatos de video, reproductores de videodiscos y otros tipos de máquinas.

Televisores domésticos
La calidad de imagen de los televisores domésticos ha ido mejorando rápidamente con los adelantos en la tecnología de la construcción de tubos de rayos catódicos y la alineación de cañones de electrones. Sony, por ejemplo, ha tomado buena nota del doble servicio a que se destinan sus aparatos al instalar el conector de la antena en la parte delantera de los mismos

Chris Stevens

Persistencia y color

Un factor a tener en cuenta en la elección de un monitor monocromático es el tipo de fósforo utilizado. Ésta es la sustancia polvorienta que reviste la cara interior de la pantalla y tiene la propiedad de ponerse incandescente al ser alcanzada por un haz de electrones, produciendo de este modo la imagen.

Las principales consideraciones atañen al color y a la "persistencia". La primera se explica por sí misma, pero la persistencia se suele comprender menos. Se trata de una medida del tiempo que dura el fósforo incandescente después de que el haz ha pasado por él. Rara vez se facilita una cifra exacta, pues por lo general basta con indicar persistencia *larga* o *corta*.

Los fósforos que se utilizan en los televisores y virtualmente en todos los monitores en color tienen una persistencia corta, al igual que la mayoría de los monitores monocromáticos, pero en muchas aplicaciones es preferible que la imagen continúe brillando durante una fracción de segundo, porque ello reduce el "pestañeo" de la pantalla, una de las principales causas de la fatiga visual.

El ejemplo más conocido del fósforo de persistencia larga es el de las pantallas de radar, en el cual el barrido radial del haz deja una larga huella de incandescencia, confiriéndole a la pantalla una especie de "memoria" sin necesidad de una complicada electrónica.

No obstante, si el sistema emplea un lápiz óptico, la persistencia larga es un riesgo; incluso lo haría inutilizable, debido a que su operatividad depende de que la imagen desaparezca con la rapidez suficiente como para que el ordenador localice el lápiz. Si el fósforo continúa incandescente, el ordenador verá luz, independientemente de dónde señale el lápiz y sin considerar tampoco si el haz está en realidad explorando ese punto o no. Por lo tanto el ordenador no será capaz de calcular la posición del lápiz óptico valiéndose de su conocimiento de la posición explorada en un momento dado.

Existe toda una gama de tipos y colores de fósforo, según la utilización a que se destine el monitor y la preferencia del comprador. Se puede producir casi cualquier color. El fósforo blanco de persistencia corta es barato y fácil de conseguir, pero uno de los tonos de verde es casi tan común como el blanco y es mucho más descansado para la vista, al igual que el ámbar. El azul se emplea en muchos terminales de unidad principal como los utilizados en las agencias de viajes y en los mostradores de las líneas aéreas, mientras que el rojo se utiliza en las salas de radar y otros lugares en los cuales la visión nocturna no debe ser perjudicada.

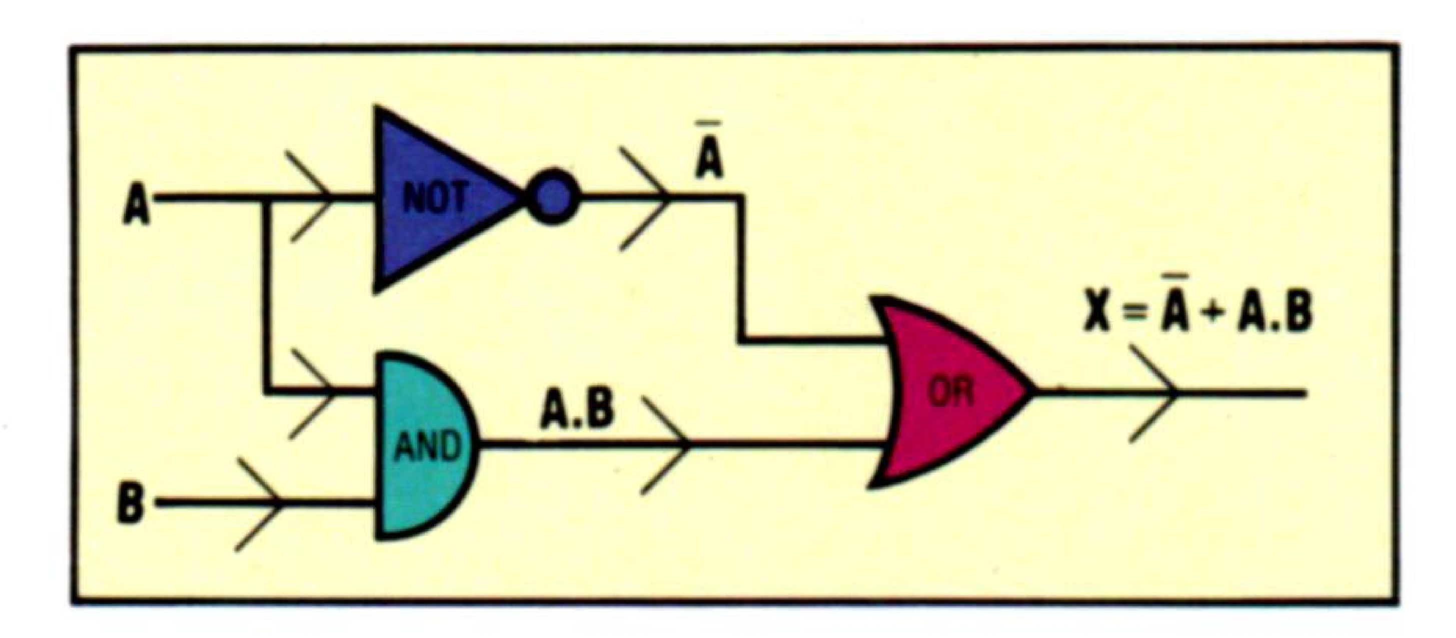
Los bloques lógicos de la suma

Veamos cómo los bloques lógicos ADN, OR y NOT se convierten en circuitos lógicos capaces de sumar

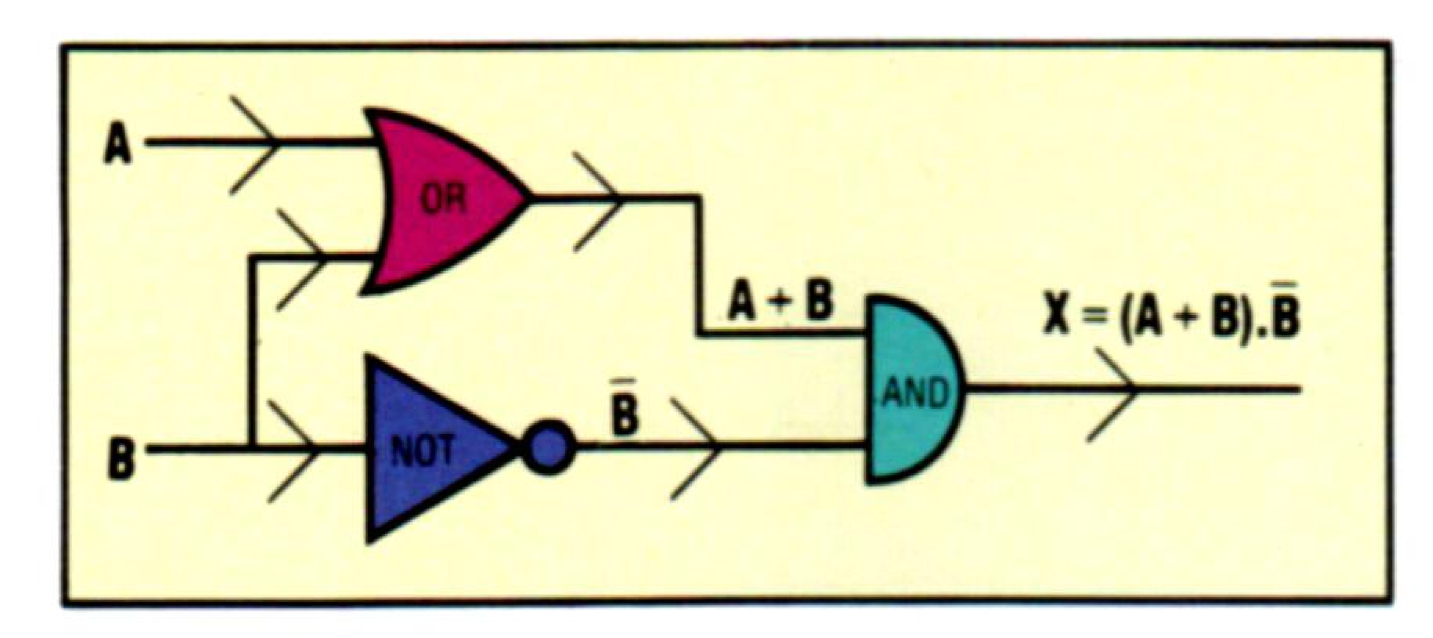
Al sistema que utiliza la notación algebraica para describir las relaciones lógicas se denomina *álgebra booleana*, en honor del lógico y matemático británico George Boole (1815-1864).

El álgebra booleana es de gran valor para el diseño de circuitos de ordenador porque permite la simplificación matemática de los circuitos lógicos. Esto significa que para realizar una función determinada se requieren menos puertas lógicas, lo que a su vez aumenta la velocidad de operación de la máquina.

Ya nos hemos encontrado con la notación booleana para la salida de las tres puertas lógicas básicas: AND (A.B), OR (A + B) y NOT ($\bar{A}$). Utilizando estas tres expresiones se pueden representar circuitos más complejos. Por ejemplo, la expresión booleana $X = \bar{A} + A.B$ representa el siguiente circuito:



Es importante observar que el orden en que se efectúan las operaciones AND y OR es decisivo. Una sencilla regla dice que AND tiene prioridad sobre OR (así como sobre NOT). Si se desea invertir el orden de prioridad, se deben utilizar paréntesis, como en este ejemplo: $X = (A + B).\bar{B}$. Para esta expresión, las dos entradas se someten a OR antes de someterse a AND con el negativo de $\bar{B}$. Éste es el diagrama del circuito:



Al dibujar un circuito lógico a partir de su expresión booleana suele ser mejor comenzar por la salida e ir trabajando hacia atrás hasta las entradas. Con este método se consiguen mejores trazados.

OR inclusivo y OR exclusivo

En inglés la partícula OR (como en castellano, o) tiene dos posibles connotaciones. De la primera ya hemos hablado: Éste OR (o) ése OR (o) ambos.

La segunda acepción tiene importantes consecuencias para el diseño de circuitos lógicos: Éste OR (o) ése, pero no ambos.

Por ejemplo, si le dicen que un amigo está siguiendo un partido de fútbol por televisión OR (o) por radio, puede entender que lo está siguiendo por ambos medios a la vez (caso de uso *inclusivo*). Por otra parte, su amigo puede ser alto OR (o) puede ser bajo, pero no ambas cosas (uso *exclusivo*).

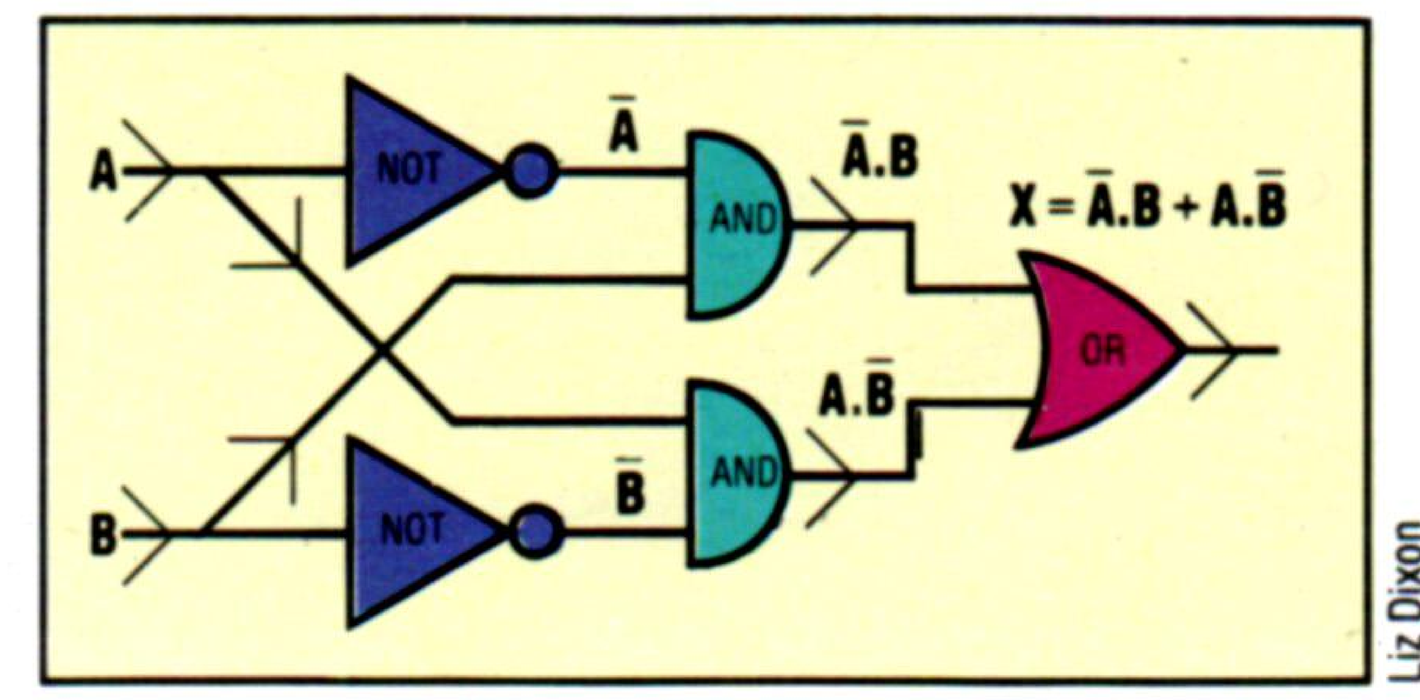
En los circuitos lógicos, la operación excluyente de OR se indica por XOR, y puede construirse a partir de juegos de puertas AND, OR y NOT. La tabla de verdad para XOR es:

ENTRADAS		SALIDAS
A	B	A ⊻ B
0	0	0
0	1	1
1	0	1
1	1	0

Como se puede ver en las filas segunda y tercera, la salida será VERDAD sólo si enfrentamos:

NOT(A) AND B
OR
A AND NOT(B)

A lo que es posible darle la siguiente expresión booleana: $X = \bar{A}.B + A.\bar{B}$. Un circuito para producir la operación *OR exclusivo* sería:



Lo que nos da un circuito de cinco puertas. Más adelante veremos cómo se puede convertir este circuito en uno de sólo cuatro puertas.



Puertas lógicas y aritmética

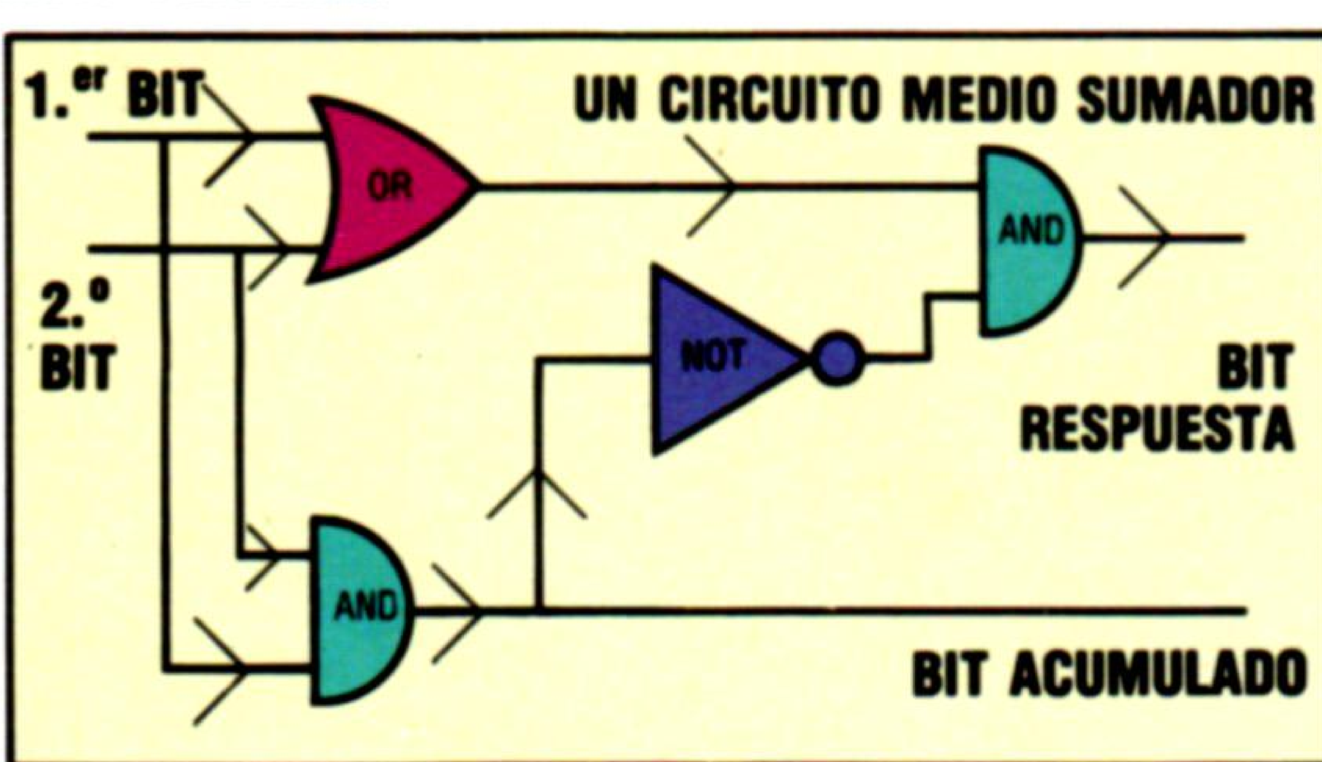
Aunque la mayoría de los ordenadores personales pueden efectuar la gama completa de operaciones aritméticas, sólo la suma se efectúa directamente mediante circuitos lógicos. Las restantes —resta, multiplicación y división— se realizan combinando hardware de circuitos "sumadores" con software para controlar el movimiento de los patrones de bits. Pero antes de que podamos analizar los circuitos que realizarán la suma binaria, hemos de echar un vistazo al proceso en sí mismo. Sea la siguiente suma binaria 101 + 111 (véase p. 55):

	8	4	2	1
	0	**1**	**0**	**1**
	0	**1**	**1**	**1**
	1	**1**	**0**	**0**
RESULTADO ACUMULADO	1	1	1	

Examinando una columna por sí misma (sea la columna de los doses) podemos listar sus posibles entradas y salidas. Las *entradas* son: los dos bits a sumar y el bit acumulado de la columna anterior. Las *salidas* son: el bit colocado como resultado en la columna de los doses, y el bit a llevar a la columna siguiente. El dispositivo capaz de aceptar estas entradas y producir las salidas correctas se llama *sumador*. Este dispositivo es bastante complejo, de modo que empecemos por una versión ligeramente simplificada, denominada *medio sumador*. Un circuito medio sumador ignora el hecho de que puede haber un uno acumulado de la columna anterior. Así se reduce el problema a un circuito con dos entradas y dos salidas. Ahora podemos elaborar una tabla de verdad para un circuito medio sumador, que adquiere la siguiente forma:

ENTRADAS: 1.er BIT	ENTRADAS: 2.° BIT	SALIDAS: BIT ACUM.	SALIDAS: BIT RESULT.
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Es fácil ver que el bit acumulado será 1 si el primer bit AND (y) el segundo bit son los dos 1. El bit resultado se forma operando con OR (o) las entradas, excepto en el caso en que el bit acumulado sea 1. Podemos decir que la salida del bit resultado es 1 si "el primer bit es 1 OR el segundo bit es 1 AND el bit acumulado es NOT 1". El circuito que incluimos a continuación producirá las salidas deseadas:



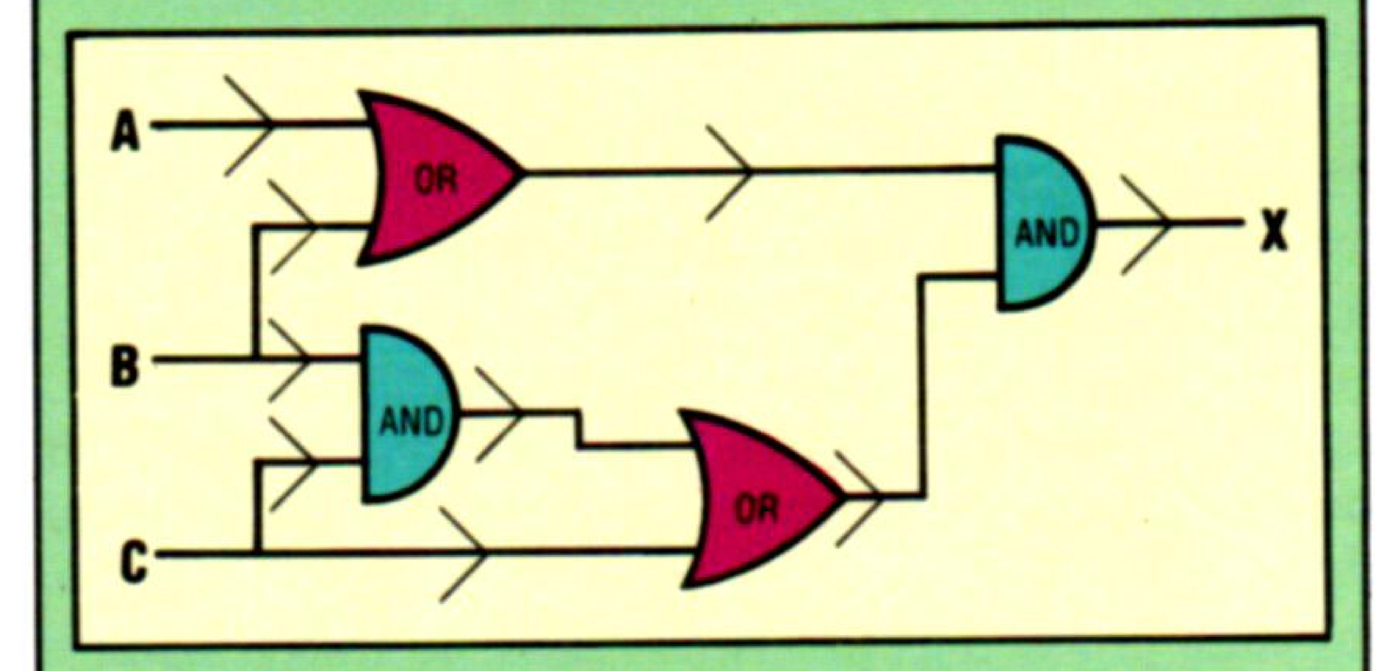
EJERCICIO 2

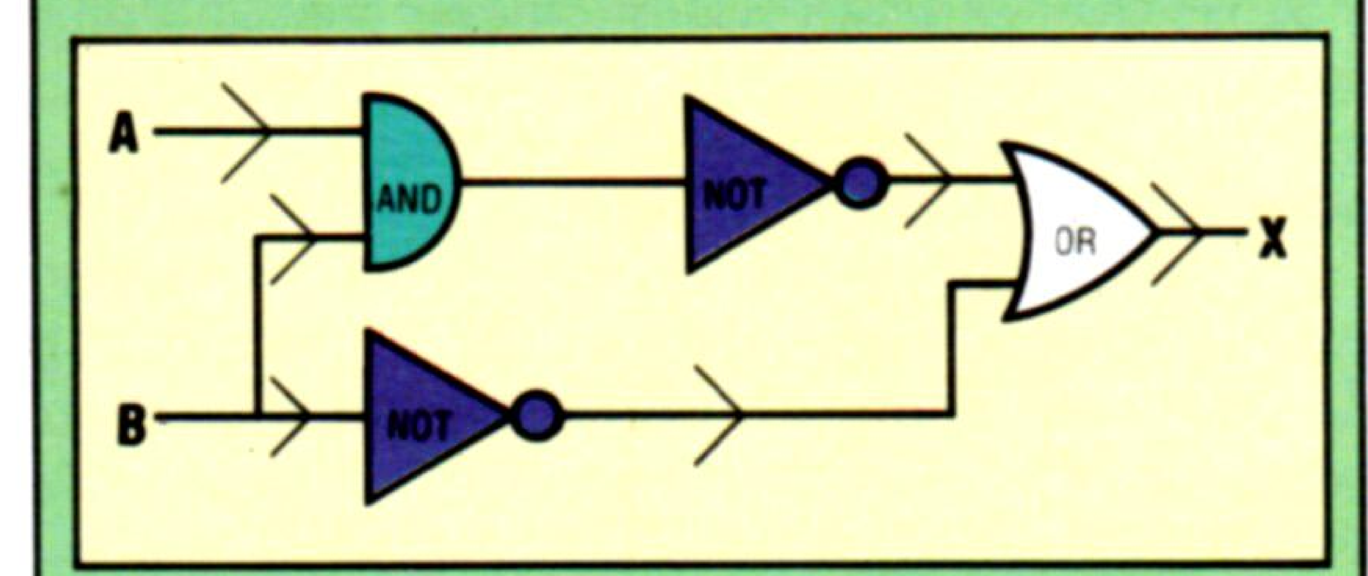
1) Dibujar los circuitos lógicos para las siguientes expresiones booleanas:

a) $X = (A + B).C$
b) $X = A.B + (A + C)$
c) $X = \overline{A}.B + (A + B)$
d) $X = \overline{A.B}.(A + B)$

2) Escribir las expresiones booleanas utilizando A y B como entradas para el bit acumulado y el bit respuesta de un circuito medio sumador.

3) Escribir las expresiones booleanas para estos circuitos:





Respuestas al ejercicio 1 de la página 489

1)

APROBADO EXAMEN CONDUCIR	ACOMPAÑADO CONDUCTOR CUALIF.	PODRÍA CONDUCIR
FALSO	FALSO	FALSO
FALSO	VERDADERO	VERDADERO
VERDADERO	FALSO	VERDADERO
VERDADERO	VERDADERO	VERDADERO

2)

DISPONE GRABADORA CASSETTE	DISPONE UNIDAD DISCO	ESCRITO PARA OTRA MÁQUINA	PROGRAMA SE PUEDE CARGAR
FALSO	FALSO	FALSO	FALSO
FALSO	FALSO	VERDAD.	FALSO
FALSO	VERDAD.	FALSO	VERDAD.
FALSO	VERDAD.	VERDAD.	FALSO
VERDAD.	FALSO	FALSO	VERDAD.
VERDAD.	FALSO	VERDAD.	FALSO
VERDAD.	VERDAD.	FALSO	VERDAD.
VERDAD.	VERDAD.	VERDAD.	FALSO

3)

A	B	P	Q	C
0	0	1	0	0
0	1	1	1	1
1	0	0	1	0
1	1	0	1	0

Rizar el rizo

Con el Spectrum se llegó al ordenador personal potente, pequeño y barato: faltaba sólo el microdrive, el dispositivo de acceso rápido

El sistema del microdrive utiliza un bucle continuo de 200 pulgadas de cinta de video magnética de 2 mm que rota en el interior de un pequeño cartucho una vez cada siete segundos. A pesar de que en cuanto a acceso e instrucciones es bastante similar a una unidad de disco flexible estándar, en realidad es una "cinta flexible". Los datos se graban digitalmente, al contrario que en el método de cinta de audio, que graba tonos. La videocinta tiene un formato de dos pistas en las cuales se graban secuencialmente los bits según un patrón en zigzag que se superpone. Este sistema, que requiere dos cabezas de lectura-escritura, permite almacenar los datos al doble de la densidad y velocidad que en un sistema de pista única. Las pistas, además, se formatean en bloques de 512 bytes. Cada bloque contiene una descripción de los datos contenidos en el mismo y está precedido por un *encabezamiento* compuesto por 26 bytes de datos de identificación.

Al conjunto de un bloque y su correspondiente encabezamiento se los denomina *sector*. El bucle de cinta de 200 pulgadas puede almacenar casi 200 sectores, dando una densidad de almacenamiento de datos de aproximadamente 500 bytes por pulgada. Un archivo identificado por su nombre se almacena en un solo sector si su longitud es de menos de 512 bytes. Si se superpone con otro sector, o con varios, sin acabar de rellenar el último sector, el espacio inutilizado de éste se pierde hasta que el archivo se borre. De modo, entonces, que a pesar de que cada microdrive teóricamente puede almacenar 100 Kbytes de datos, en realidad su capacidad es de 85 a 90 Kbytes. El tiempo de acceso promedio para hallar y cargar un programa en la memoria del Spectrum es de entre 10 y 15 segundos.

La interface ZX

El microdrive se conecta al Spectrum a través de una interface 1 ZX, que se acopla en el conector marginal para periféricos del micro. Se pueden conectar hasta ocho microdrives en la disposición de "cadena margarita". Además de proporcionar una conexión para microdrives, la interface 1 sirve como interface RS232 estándar, como interface para red de área local y como conector para una impresora ZX. También contiene instrucciones en ROM que amplían el BASIC de Sinclair al objeto de

Identificación de archivos

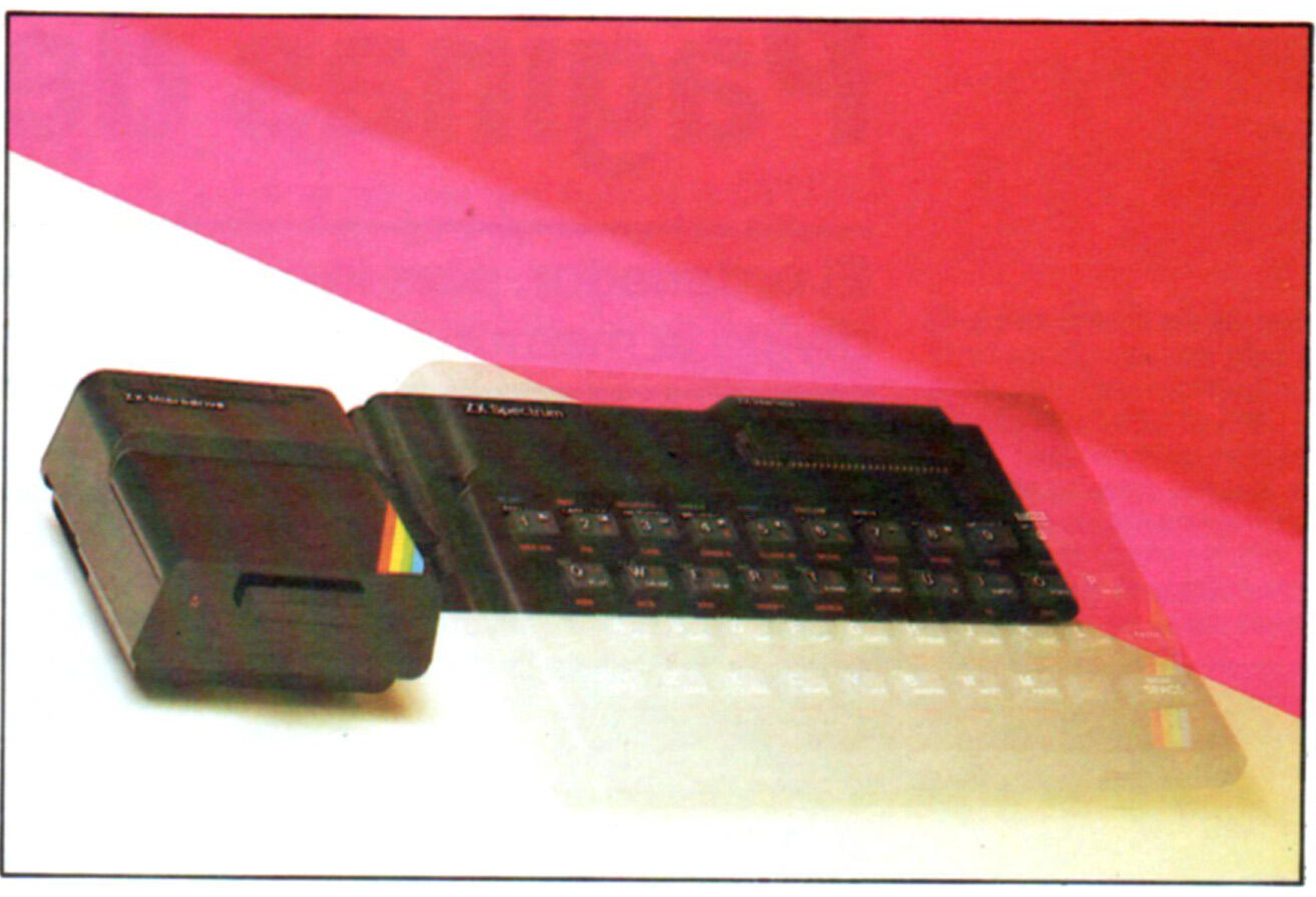
Los programas y los archivos de datos se escriben en el mismo formato: un archivo en cinta comienza con un bloque de encabezamiento que contiene el nombre del archivo y el número de bloques de datos que hay en éste; luego se escriben los datos del usuario, en bloques de 512 bytes, a cada uno de los cuales se les antepone información de sincronización, de identificación y de verificación de errores que proporciona el microprocesador

Los bytes de datos se escriben en la cinta en dos pistas: el primer byte se escribe en la pista 1, el segundo byte se escribe en la 2, desplazado medio byte respecto a la línea de comienzo del primer byte; el tercer byte se escribe en la pista 1 y al lado del primer byte, el cuarto byte se escribe en la pista 2 al lado del segundo byte, y así sucesivamente

Alun Jones

incluir órdenes para manipulación de datos adecuados al microdrive y otras interfaces.

El microdrive ZX y la interface 1 son eficaces accesorios para el sistema Spectrum. Las facilidades que ofrecen reproducen con mucha fidelidad la operación de un sistema de disco flexible convencional. Sin embargo, existen las limitaciones impuestas por la incierta fiabilidad a largo plazo de los cartuchos de cinta y la falta de una auténtica facilidad para archivos de acceso directo. Esto afecta la viabilidad del microdrive como sistema de almacenamiento para aplicaciones serias de gestión empresarial, porque la mayoría del software comercial requiere la localización, el almacenamiento y la recuperación de pequeñas cantidades de datos muchas veces durante su utilización. El advenimiento del ordenador de gestión Sinclair QL, que se suministra con dos microdrives incorporados, bien podría ser el origen de una ampliación de las capacidades del sistema.



Ian Mckinnell

Cinta flexible
Combinando el bajo precio y la simplicidad del almacenamiento en cinta con la velocidad del disco, el microdrive se ha llegado a conocer como una cinta flexible. Se conecta con el Spectrum a través de la interface 1.
Conectores de sonido para acceso a cassette y una puerta RS232 son también configuraciones de la interface 1

Órdenes del microdrive

Las órdenes que se utilizan con el microdrive son:

FORMAT CAT SAVE* VERIFY* LOAD*
MERGE* ERASE OPEN# PRINT# INPUT#
INKEY$# CLOSE# MOVE

En todos los casos, M selecciona el microdrive, N es el número del microdrive al que acceder (1-8) y S es el número de flujo asignado (4-15).

Format
Formatea la configuración de los datos en cinta, da nombre al cartucho y borra todos los datos grabados previamente. Se puede construir de la siguiente manera:

```
FORMAT "M";N;"NAME" o
FORMAT M$;N;C$
```

donde NAME es el nombre seleccionado para el cartucho (1-10 caracteres), M$ (ya sea M o m) es lo mismo que M y C$ (1-10 caracteres) es como NAME.

Cat
Esta orden carga a la pantalla o flujo especificado un catálogo de todos los archivos contenidos en cartucho en una unidad determinada. Se obtiene con:

```
CAT N o
CAT# S;N
```

El catálogo contiene el nombre del cartucho, hasta 50 nombres de archivo y el espacio libre en el cartucho expresado en Kbytes.

Save*
Crea archivos de programas que pueden ser programas, cadenas o datos con nombre, y se construye de una de las siguientes maneras:

```
1. SAVE* "M";N;"NOMARCHI"
2. SAVE* "M";N;"NOMARCHI"SCREEN$
3. SAVE* "M";N;"NOMARCHI"DATA A()
4. SAVE* "M";N;"NOMARCHI"LINE X
```

creando lo siguiente:

1. Un archivo
2. Un archivo que conste de SCREEN$
3. Un archivo que conste de datos A()
4. Un archivo que se ejecutará (RUN) desde la línea X al cargarlo (LOAD).

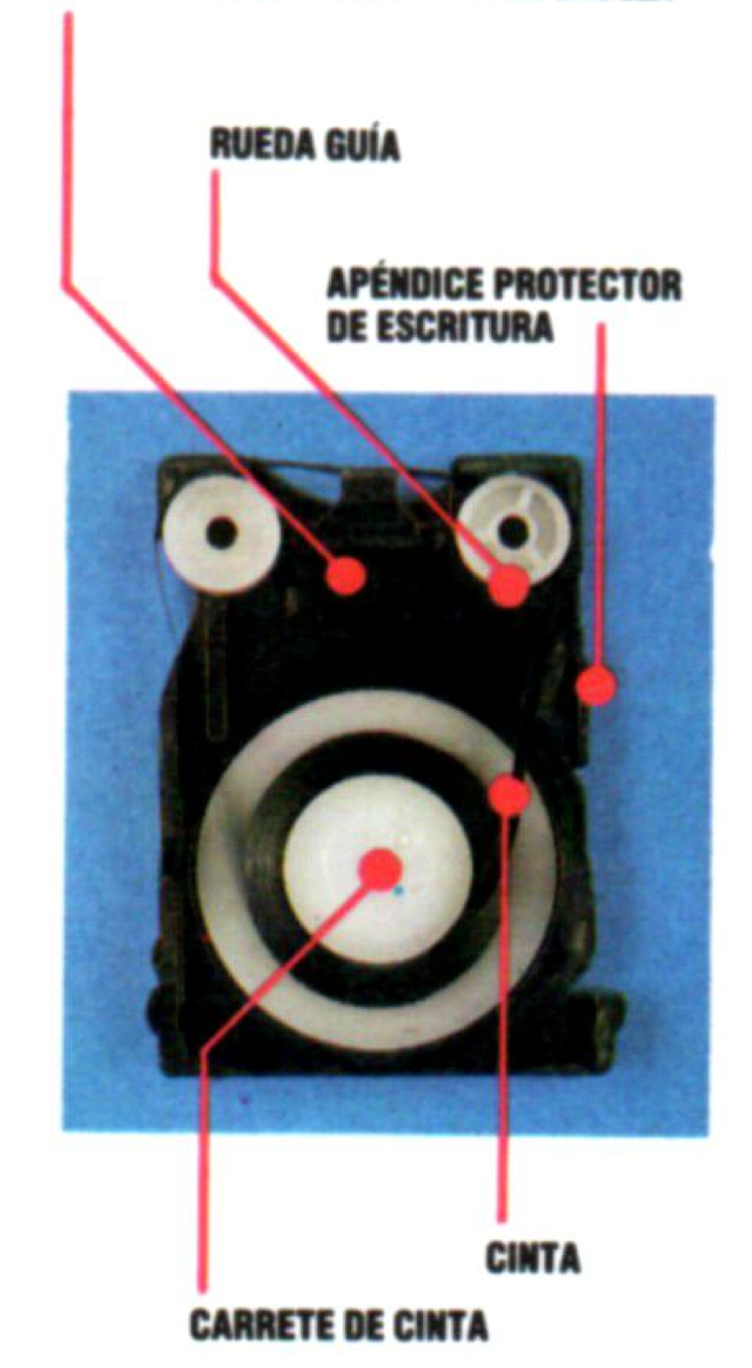
Verify*, Merge* y Erase*
Estas órdenes se construyen de la misma forma que 1. SAVE* anterior. VERIFY* compara el archivo "NOMARCHI" con el contenido corriente de la memoria para el usuario y genera un mensaje de error si son diferentes. MERGE* une "NOMARCHI" con el contenido corriente de la memoria para el usuario, y ERASE* borra "NOMARCHI".

Load*
LOAD* se puede construir de la misma forma que 1. SAVE* y 2. SAVE*. Al ejecutarla, LOAD* copia el contenido del archivo especificado en la memoria para el usuario.

Open#, Print#, INPUT#, Inkey$#, Close# y Move
Estas órdenes están relacionadas con la manipulación de archivos de datos. Éstos se almacenan secuencialmente, pero se pueden manipular para que imiten algunas de las propiedades de los archivos de acceso directo permitiendo leer un archivo de datos entero y extrayendo los datos requeridos después de cargarlo. Del mismo modo, ese archivo se puede modificar y realmacenar. La manipulación de archivos de datos se organiza abriendo y cerrando (OPEN y CLOSE) flujos para establecer canales de datos haciendo referencia al número de flujo especificado. Por ejemplo:

```
OPEN#S;"M";N;"NOMARCHI"
```

conecta el flujo S con el archivo "NOMARCHI" en el microdrive N. Se puede, entonces, escribir en "NOMARCHI" utilizando PRINT#S y leerlo empleando INPUT#S o INKEY$#S. MOVE se puede utilizar para transferir archivos de datos dentro de un cartucho, de un microdrive a otro, o a cualquier dispositivo al cual se pueda acceder a través de un número de flujo. Cuando los canales de flujo ya no se necesitan, se desconectan mediante CLOSE#S.



Wafer del microdrive
La cinta del wafer es una videocinta de 2 mm, utilizada en virtud de su fortaleza y su elevada densidad de almacenamiento. La cinta forma un bucle de 8 metros y está revestida, de modo que se desliza con facilidad desde el carrete de cinta central a la cinta rebobinada. Al estar en su lugar un apéndice protector de escritura permite leer datos del wafer o escribirlos en él; si se quita el apéndice, el wafer sólo se puede leer, protegiendo, por tanto, la información almacenada

Páginas de memoria

El ordenador, para localizar de manera expedita cualquier byte, divide la memoria en "páginas", como si se tratara de un libro

En el primer capítulo de esta serie sobre el lenguaje máquina vimos por analogía la forma en que los ordenadores almacenan la información con la corriente eléctrica. Utilizamos el ejemplo de una fábrica en la que cada trabajador tenía un modelo propio de interruptor, que encendía cuatro bombillas en el despacho del director, lo que permitía identificar quién era el empleado que había llegado al trabajo.

En nuestro ejemplo descubrimos que utilizando cuatro interruptores y bombillas podíamos representar los números entre 0 y 15. En otras palabras, sólo había 16 modelos posibles. Pero, en cambio, si hubiéramos empleado ocho interruptores y ocho bombillas, habríamos dispuesto de 256 modelos exclusivos y, por consiguiente, hubiésemos podido contar desde 0 hasta 255 (variaciones de 2 elementos, OFF y ON, en grupos de 8 cada vez = $= 2^8 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 256$).

Pues bien, un ordenador personal tiene la memoria dispuesta en bancos individuales de ocho interruptores y cada uno de estos bancos se denomina *byte*. En general, la CPU manipula la información de un byte por vez, lo que de hecho significa que sólo puede sumar, comparar y almacenar números entre 0 y 255. Podría parecer que esto limita sus capacidades aritméticas, pero no es así. Piense, por ejemplo, cómo hace una suma cualquiera, 63951 + 48770 = ?, y verá que en realidad manipula los dígitos uno a uno. Del mismo modo, la CPU puede efectuar sumas de números muy grandes utilizando un byte cada vez.

Debido a que posee ocho interruptores, un byte es un lugar donde se puede almacenar un número binario de ocho dígitos. Cada una de estas posiciones de dígitos binarios se denomina *bit*. El bit de un byte está o encendido (ON) o apagado (OFF) y un dígito binario o es un 1 o es un 0.

Suele ser importante individualizar cada uno de los bits de un byte, de modo que por convenio los bits se numeran de 0 a 7 de derecha a izquierda del byte. Si un byte contiene el número binario 00000001, decimos entonces que el bit 0 es 1, o que el bit 0 está ON, o que el bit 0 está ACTIVADO; todos los otros bits son 0, o están OFF, o están DESACTIVADOS. De modo que, por cuanto concierne al número binario 01001000: el bit 3 está ACTIVADO, al igual que el bit 6, el bit 4 está OFF, el bit 7 es 0, el bit 0 está DESACTIVADO, etc. En un byte, también se suele decir que el bit 0 es el bit menos significativo (LSB: *Least Significant Bit*) y el bit 7 el bit más significativo (MSB: *Most Significant Bit*).

Podemos, pues, concebir la memoria del ordenador como una larga tira de papel cuadriculado, de ocho cuadrados de ancho y miles de cuadrados de largo: cada fila de ocho cuadrados es un byte, y esos mismos cuadrados son los bits de ese byte. La memoria no sirve de nada si no se pueden posicionar datos en ella, de modo que cada uno de los bytes posee una etiqueta de identificación que se denomina *dirección*; la dirección de un byte no está escrita en ningún lugar del papel (o del byte), sino que sencillamente es el número del byte en la memoria, contando a partir del principio de la misma. El primer byte, por lo tanto, recibe la dirección 0, el byte siguiente la dirección 1, el siguiente la dirección 2, y así sucesivamente. Si se desea escribir algo en el byte 43, contaremos los bytes, empezando por la parte inferior de la memoria (por el byte 0), hasta llegar al byte 43.

Una vez alcanzado, no habrá nada que identifique a ese byte como el byte 43, excepto su posición: se ha ido contando hacia arriba a partir del byte 0 y así llegamos al 43, de modo que ése ha de ser el byte 43. Los bytes de la memoria son en realidad minúsculos bancos de dispositivos de ocho transistores grabados en los chips del interior de la máquina y son idénticos entre sí en todo, excepto en su posición física.

Sin embargo, este método tiene un inconveniente. Este sistema de direccionamiento de memoria sería apropiado si sólo hubiera unos pocos centenares de bytes. La CPU puede contar de 0 a 100 en fracciones de milisegundo; pero los ordenadores poseen millares de bytes, y contar de 0 a 20 000 ocupa un tiempo apreciable, aun para un microprocesador. El ordenador tiene una forma de superar este problema y consiste en dividir la memoria en páginas, al igual que sucede con los libros.

Si seguimos pensando en la memoria de un ordenador como una tira de papel cuadriculado de millares de cuadrados de longitud y ocho cuadrados de anchura, podemos imaginariamente cortar esa tira cada 100 bytes (o sea, cortar por la línea que divide el byte 99 y el byte 100, cortar de nuevo por la que separa el byte 199 y el byte 200, por el byte 299 y el byte 300, y así sucesivamente). Ahora cada una de las tiras de papel entre los cortes es una página de 100 bytes. La página 0 empieza en el byte 0 y sigue hasta el byte 99; la página 1 empieza en el byte 100 y sigue hasta el byte 199; la página 2 comprende desde el byte 200 hasta el byte 299, etc. Según esto, para hallar ahora cualquier byte, supongamos el byte 3 518, no necesitamos contar 3 518 bytes desde el principio de la memoria, porque la dirección nos indica que este byte debe estar en la página 35. En consecuencia, sólo necesitamos contar 35 páginas desde la parte inferior de la memoria y después contar los bytes desde la parte inferior de esa página hasta llegar al byte 18 de la misma, que debe ser el byte 3 518. Pruebe, para verlo mejor, con una tira de papel cuadriculado.

Este sistema de memoria con paginación es conveniente porque podemos tomar la dirección de cualquier byte y dividirla en dos partes: los dígitos



desde la columna de las centenas hacia la izquierda corresponden al número de página del byte, y los dígitos desde la columna de las decenas hacia la derecha representa el número de bytes contados desde la parte inferior de esa página. En el ejemplo anterior separamos la dirección 3 518 en dos números: el número de página 35 y el número de byte 18 de esa página. Decimos que 18 es un *offset* (desplazamiento) o un *page offset* (desplazamiento de página), porque es el número en razón del cual uno ha de desplazar (o incrementar) la dirección del byte inferior hasta el byte en cuestión.

El ordenador, no obstante, no cuenta en decimal, como contamos nosotros, sino que cuenta en binario. El sistema de paginación consiste en ser capaz de hallar la página y el desplazamiento sencillamente mediante la inspección de la dirección del byte. Ya vimos que el decimal 99 se representa 01100011 en binario, y el decimal 100 es el binario 01100100; el decimal 199 es 11000111, y el decimal 200 es el binario 11001000. A partir de estos ejemplos podemos apreciar que no existe una forma sencilla de saber qué página es, con sólo observar los números binarios, como hacemos tan fácilmente con los equivalentes decimales.

Hemos elegido 100 como tamaño de la página porque se trata de un número significativo en el sistema decimal (es una potencia de 10). Sin embargo, si hemos de contar en binario, entonces debemos elegir un tamaño de página que sea adecuado para *ese* sistema. El tamaño de página que utilizan nuestros ordenadores es 256, de modo que la página 0 empieza con el byte 0 y continúa hasta el byte 255; la página 1 empieza con el byte 256 y continúa hasta el byte 511, etc. Para ver la conveniencia de esta nueva numeración de las páginas debemos escribir las direcciones en binario:

Página 0: byte 00000000 – byte 11111111
Página 1: byte 100000000 – byte 111111111

Como apreciará, podemos contar en binario de 0 a 255 en números de ocho dígitos; el número siguiente (256) requiere nueve bits, y con nueve bits podemos contar hasta 511. El número siguiente (512) requiere diez bits, y con diez bits podemos contar hasta 1023; y así sucesivamente. Ahora vemos que si el tamaño de la página es 256 y contamos en binario, un desplazamiento sobre la página sólo puede ser indicado por los ocho bits situados a la derecha, pues el número de página viene dado por los bits situados a partir del bit 8 hacia la izquierda.

Esto puede resultar sorprendente dado que antes hemos afirmado que la CPU sólo puede manipular bytes individuales, y un byte sólo contiene ocho bits. Por consiguiente, puede que se esté preguntando qué sentido tiene hablar de números de nueve o diez bits. La respuesta es que todas las direcciones de la memoria se tratan como números de dos bytes y la CPU se ocupa de ellas a un byte por vez. Si volvemos a escribir los límites de páginas en forma de números de dos bytes veremos cómo este sistema nos resulta más claro:

Página 0	empieza en	00000000	00000000
	termina en	00000000	11111111
Página 1	empieza en	00000001	00000000
	termina en	00000001	11111111
Página 10	empieza en	00000010	00000000
	termina en	00000010	11111111
Página 11	empieza en	00000011	00000000
	termina en	00000011	11111111

y así sucesivamente.

Ahora podemos apreciar que cuando la CPU recoge información de un byte de la memoria, o cuando coloca información en él, dicho byte se puede identificar mediante una dirección de dos bytes. El primero de los dos bytes, o el situado más hacia la izquierda, da el número de página, mientras que el segundo byte, o el situado más a la derecha, da el desplazamiento.

Al final de esta sección proporcionamos programas que convierten decimales a binarios y a hexadecimales.

Direccionamiento por páginas
El direccionamiento por páginas divide a la memoria en bloques o páginas imaginarias de 256 bytes. Todas las direcciones se expresan así como números de dos bytes: un byte proporciona el número de página y el otro proporciona el desplazamiento desde el principio de esa página

Procesador de números

Los tres programas que presentamos aquí, para el BBC Micro, Spectrum y los Commodore, reciben números decimales y devuelven sus equivalentes binarios y hexadecimales

Commodore 64

```
10 REM*******COMMODORE**********
40 S$="=         ":X$=
   "0123456789ABCDEF"
50 REM  S$ CONTIENE 9 ESPACIOS
60 PRINT CHR$(147)     :REM LIMPIAR
   PANTALLA
70 PRINT "VISUALIZAR NUMEROS
   DECIMALES"
80 PRINT "Y SUS EQUIVALENTES
   BINARIOS"
90 PRINT:PRINT "    ******DIGITE
   EL 0 PARA SALIR********":PRINT
100 FOR K=1 TO 1
110 FOR L=1 TO 1
120 INPUT"DIGITE CUALQUIER NUMERO
    ENTERO POSITIVO";A$
130 NU=VAL(A$)
140 IF NU=0 THEN PRINT"SALIDA
    PROGRAMA":STOP
150 IF INT(NU)<>ABS(NU) THEN L=0
160 IF NU>65535 THEN PRINT NU;"ES
    DEMASIADO GRANDE":L=0
170 NEXT L
200 NM=NU:H$="":GOSUB 2000
210 PRINT NU;TAB(5);N$;
220 IF RIGTH$(A$,1)="+" THEN
    GOSUB 4000
230 PRINT H$:PRINT:PRINT
240 K=0:NEXT K
250 END
300 END
1000 REM******S/R BYTE
     BINARIO******
1010 B$=""
1020 FOR D=8 TO 1 STEP-1
1030 N1=INT(N/2)
1040 R=N-2*N1
1050 B$=MID$(STR$(R),2)+B$
1060 N=N1
1070 NEXT D
1080 RETURN
2000 REM****S/R CONVERSION
     BINARIO***
2010 IF NM<256 THEN N=NM: GOSUB
     1000:N$=S$+B$:RETURN
2020 HI=INT(NM/256):LO=NM-256*HI
2030 N=HI:GOSUB 1000:N$="= "+B$
2040 N=LO:GOSUB 1000:N$=N$+" "+B$
2050 RETURN
3000 REM****S/R BYTE
     HEXA*********
3010 HB=INT(N/16):LB=N-HB*16
3020 B$=MID$(X$,HB+1,1)+MID$
     (X$,LB+1,1)
3030 RETURN
4000 REM****S/R CONVERSION
     HEXA******
4010 IF NM<256 THEN N=NM:GOSUB
     3000:H$="=  "+B$: RETURN
4020 HI=INT(NM/256):LO=NM-256*HI
4030 N=HI:GOSUB 3000:H$="= "+B$
4040 N=LO:GOSUB 3000:H$=H$+" "+B$
4050 RETURN
```

BBC Micro

Copiar el listado del Commodore con las siguientes modificaciones:

```
60 CLS:@%=5
210 PRINT TAB(0);NU;TAB(5);N$;
1050 B$=STR$(R)+B$
```

Este programa no hace uso de las facilidades para representación de números del BBC para mantener la compatibilidad de formato con las otras máquinas: quizá usted sea capaz de volver a escribirlo en una versión más corta.

Spectrum

```
10 REM*******SPECTRUM********
40 LET S$="=         ":LET X$=
   "0123456789ABCDEF"
50 REM   S$ CONTIENE 9 ESPACIOS
60 CLS
70 PRINT "VISUALIZAR NUMEROS
   DECIMALES"
80 PRINT "Y SUS EQUIVALENTES
   BINARIOS"
90 PRINT:PRINT "*******DIGITE EL
   0 PARA SALIR********":PRINT
100 FOR K=1 TO 1
110 FOR L=1 TO 1
120 INPUT"DIGITE CUALQUIER NUMERO
    ENTERO POSITIVO";A$
130 LET NU=VAL(A$)
140 IF NU=0 THEN PRINT "SALIDA
    PROGRAMA":STOP
150 IF INT(NU)<>ABS(NU) THEN LET L=0
160 IF NU>65535 THEN PRINT NU;"ES
    DEMASIADO GRANDE":LET L=0
170 NEXT L
200 LET NM=NU:LET H$="":GOSUB 2000
210 PRINT NU;TAB(5);N$;
220 IF A$(LEN A$)="+" THEN GOSUB
    4000
230 PRINT H$:PRINT:PRINT
240 LET K=0:NEXT K
300 STOP
1000 REM**S/R BYTE BINARIO**
1010 LET B$=""
1020 FOR D=8 TO 1 STEP-1
1030 LET N1=INT(N/2)
1040 LET R=N-2*N1
1050 LET B$=STR$(R)+B$
1060 LET N=N1
1070 NEXT D
1080 RETURN
2000 REM**S/R CONVERSION BINARIO**
2010 IF NM<256 THEN LET N=NM:GOSUB
     1000:LET N$=S$+B$:RETURN
2020 LET HI=INT(NM/256):LET
     LO=NM-256*HI
2030 LET N=HI:GOSUB 1000:LET
     N$="= "+B$
2040 LET N=LO:GOSUB 1000:LET
     N$=N$+" "+B$
2050 RETURN
3000 REM****S/R BYTE HEXA******
3010 LET HB=INT(N/16):LET
     LB=N-HB*16
3020 LET B$=X$(HB+1)+X$(LB+1)
3030 RETURN
4000 REM***S/R CONVERSION HEXA****
4010 IF NM<256 THEN LET N=NM:GOSUB
     3000:LET H$="=  "+B$:RETURN
4020 LET HI=INT(NM/256):LET
     LO=NM-256*HI
4030 LET N=HI:GOSUB 3000:LET
     H$="= "+B$
4040 LET N=LO:GOSUB 3000:LET
     H$=H$+" "+B$
4050 RETURN
```

Si se da entrada a un numero con un "+" al final, por ejemplo: 6435+, entonces, además de su representación decimal y binaria, obtendrá también su representación hexadecimal



La reina del juego

Al igual que otras grandes firmas, Atari surgió al cristalizar una idea: Nolan Bushnell, al inventar el juego llamado "Pong", no imaginó que aquello era apenas el principio...

El sencillo método de Nolan Bushnell de poner el control de lo que aparecía en la pantalla en las manos de quien lo estaba contemplando, habría de transformar el concepto popular del ocio y cautivar a millones de jóvenes.

Bushnell y sus dos socios, Ted Dabney y Larry Bryan, aportaron cada uno 100 libras esterlinas para lanzar el Pong. El juego hizo su primera aparición en Sunnyvale, California, en 1972, y pronto hubo pruebas de que el invento iba a ser algo sensacional y rentable. La hegemonía de Atari sobre el mercado de videojuegos comenzó con una astuta decisión tomada al poco tiempo: la de comprar los derechos a Bushnell.

Atari se mantuvo a la cabeza del mercado durante la mayor parte de la década de los setenta, hasta que el gusto del público pasó de las máquinas de juegos recreativos a los micros personales. Comercializar juegos es como comercializar discos: uno debe descubrir sus potenciales estrellas "pop" y promocionarlas. Es bastante normal, en consecuencia, que Atari sea propiedad de la multinacional Warner Comunications International, conocida entre nosotros por sus intereses en las industrias del cine y discográfica. Y aunque los negocios de Atari en máquinas recreativas tragaperras reportaron abundantes beneficios a finales de los setenta, en 1983 los ingresos disminuyeron un 25 %, lo que ocasionó cuantiosas pérdidas a la empresa madre.

El juego *Space invaders* (Invasores del espacio) de Taito, que Atari ha comercializado sabiamente, es el más conocido de todos los juegos por ordenador. Se convirtió en un fenómeno social y dio origen a todo un universo de juegos de persecución intergaláctica. A finales de los años setenta Atari estaba en el centro del *boom* de los juegos recreativos. La empresa seguía produciendo éxitos uno tras otro: *Asteroides, Battlezone, Ciempiés, Lunar lander, Comando de misiles* y *The tempest*.

Pero el *boom* recreativo se extinguió con la misma vertiginosa rapidez con que había comenzado. Los clientes se volcaron a los ordenadores personales porque éstos ofrecían dos principales ventajas. Con ellos uno podía jugar a los juegos recreativos sin acudir a los salones de juegos y sin tener que apilar monedas, al mismo tiempo, uno disponía de una máquina electrónica muy flexible.

En un primer momento Atari respondió a este cambio en la demanda del mercado convirtiendo su mejor software recreativo en juegos por ordenador. Éstos utilizaban cartuchos en estado sólido que se enchufaban en la parte posterior de una unidad de ordenador personal y o bien se sumaban a la propia ROM del ordenador o la sustituían. Aunque ésta resultó ser una forma efectiva de adquirir un juego por ordenador, puesto que no requería que el jugador cargara el programa del juego en la memoria desde cassette o disco, los componentes en estado sólido hacían que los cartuchos resultaran muy costosos. Y como estos cartuchos no eran reprogramables (los programas estaban grabados físicamente en los circuitos), a menudo la empresa se quedaba con montañas de material electrónico de desecho proveniente de aquellos juegos que no conseguían cobrar la popularidad suficiente.

Los éxitos disminuyen

La línea comercial seguida por Atari pronto evidenció signos de debilitamiento. La empresa basó la cifra de futuras ventas para algunos de sus cartuchos de juegos en las de PacMan, el juego que obtuvo un éxito tan fenomenal, y al fin hubo de pagar el precio por este error de cálculo. Se hizo un inventario de los cartuchos invendibles, y 14 camiones cargaron con ellos acabando por consignarlos en un gran pozo del desierto de Nevada.

Atari tampoco se resarció con una configuración exclusiva de los productos para juegos por ordenador: convertir el código de ordenador en una entidad física no lo es todo para distribuirlo eficazmente. Se puede transmitir por teléfono o por cable, o emitir por radio o por televisión. Cada vez más están saliendo al mercado nuevas técnicas y productos que permiten estas formas de transmisión. En 1983, por ejemplo, la Romox Corporation de Estados Unidos presentó una máquina a la que llamaron *Romox Programming Terminal*. Se trataba de una máquina de discos rígidos de 15 Megabytes que se podía comunicar, a través de las líneas telefónicas, con una base de datos de software para



Cortesía de Atari

Nolan Bushnell
Los éxitos de Atari se basaron en los esfuerzos y la dedicación de un hombre: Nolan Bushnell. Cuando Bushnell creó el Pong (el primer juego para ordenadores), en 1971, sin duda ignoraba el contenido de esa caja de Pandora que estaba abriendo



Sistema exclusivo para juegos
Por muy poco dinero, el Video Computer System de Atari viene completo con dos palancas de mando, adaptador de corriente y un cartucho del juego PacMan. Al ser "exclusivo" para juegos, el VCS no se puede utilizar como ordenador para aplicaciones generales, y todos los juegos se venden en cartucho

juegos. Además, también era un dispositivo EPROM con ranuras para aceptar los conectores de los principales cartuchos. La importancia de esta máquina reside en el hecho de que ahora se puede acudir a un comerciante de la zona, disponer de una lectura instantánea de los "Veinte mejores", los veinte juegos más vendidos, y seleccionar uno de ellos para llevarlo de inmediato en un cartucho Romox en blanco.

Una alternativa a este método de distribución fue Gameline, el sistema de "paga lo que juegues" que creó Bill van Meister en Estados Unidos. Gameline comercializa un modem enchufable para máquinas Atari VCS que acopla el ordenador personal al sistema telefónico. Originalmente estos juegos cuestan un dólar por 45 minutos de juego.

Dos de las redes más grandes de informática personal de Estados Unidos, Compuserve y The Source, ofrecen software para juegos como parte de su servicio regular para proveedores que, de lo contrario, enchufan ordenadores personales en una base de datos remota a través de un modem y de la red telefónica. Coleco, una empresa fabricante de videojuegos domésticos, se ha unido con la AT&T (American Telephone and Telegraph) para proporcionar un servicio de entretenimiento interactivo. Atari entiende que éste es el camino a seguir y se ha unido a Activision para enviar programas de juegos a través de la red telefónica en su plan secreto *Ataritel*. Siempre que Atari permanezca en sociedad con Warner Communications, la cosa funcionará; pero si Atari se vendiera entonces perdería acceso a Warner Amex Cable Communications, de la cual depende el plan.



"Major Havoc"
Durante la mayor parte de la década de los setenta, Atari funcionó principalmente gracias a los beneficios que generaban las máquinas recreativas como esta que vemos en la fotografía, uno de los muchos juegos de "captura". El advenimiento del ordenador personal exigió una estrategia comercial completamente nueva

Atari ha tenido sus problemas, originados parcialmente en el pasado por la rivalidad entre la decadente sección videojuegos y el creciente departamento ordenadores personales de la empresa, y los ha resuelto ahora mediante fusión. Pero aún posee una buena gama de máquinas personales que durante mucho tiempo han sido la vanguardia de los gráficos y el software fácil de utilizar para ordenadores personales. Las máquinas personales más nuevas son similares, en cuanto a diseño, a las antiguas, que fueron notablemente avanzadas para su época. Estas máquinas configuran tres chips a la medida: Pokey, Antic y GTIA, que controlan puertas de entrada-salida, gráficos y color.

Todas las máquinas se basan en el procesador 6502 y existe en la actualidad una útil variedad de programas de utilidades. Entre éstos se incluyen: VisiCalc, Atariwriter (un paquete para tratamiento de textos) y un programa para administración doméstica. La softcard Z80 es ahora una realidad, y esto hace que los ordenadores Atari sean aptos para equiparlos con el Personal CP/M de Digital Research. La empresa también está prodigando sus atenciones al software. Su sección de Gran Bretaña ha nombrado a un mediador de software para solventar sus problemas relacionados con el software, para comercializar su mejor software convenientemente adaptado a otros micros (en particular las máquinas Commodore) y, lo más significativo, para observar de cerca el escenario británico y detectar a los jóvenes programadores de juegos. Al cabo de un par de años en los cuales los hados no le fueron muy propicios, Atari posee el potencial suficiente para recobrar sus días de esplendor.

Editorial Delta, S.A.



10026

9 788485 822836